

# **Ketterä ja asiakaslähtöinen palvelukehitys tietoliikenneteollisuudessa**

Tommi Luhtala

## **Sähkötekniikan korkeakoulu**

Diplomityö, joka on jätetty opinnäytteenä tarkastettavaksi  
diplomi-insinöörin tutkintoa varten Espoossa 15.5.2016.

**Työn valvoja:**

Prof. Raimo Kantola

**Työn ohjaaja:**

KTM DI Jani Koskinen



**Aalto-yliopisto**  
**Sähkötekniikan**  
**korkeakoulu**

Tekijä: Tommi Luhtala

Työn nimi: Ketterä ja asiakaslähtöinen palvelukehitys tietoliikenneteollisuudessa

Päivämäärä: 15.5.2016

Kieli: Suomi

Sivumäärä: 6+62

Tietoliikennetekniikan laitos

Professori: Tietoverkkotalous

Työn valvoja: Prof. Raimo Kantola

Työn ohjaaja: KTM DI Jani Koskinen

## TUTKIMUKSEN TAVOITTEET

Tässä tutkimuksessa tavoitteena oli löytää hyötyjä eri palvelukehitysmalleista Soneran palvelukehitykseen. Tutkimuksessa paneuduttiin nykyisen palvelukehitysketjun kaikkiin osa-alueisiin. Tavoitteena oli löytää alan viimeisimmät kehitysmenetelmät ja suhteuttaa niitä nykyisiin menetelmiin. Työssä tutkittiin miten palvelukehitystä tulisi johtaa ja minkälaisia vaatimuksia nämä uudet kehitysmenetelmät vaativat johtajilta.

## AINEISTO

Tutkimusta varten tehtiin 14 haastattelua, joihin osallistui henkilöitä eri yksiköistä. Kolmesta palvelukehitykseen liittyvästä yksiköstä haastateltiin neljää henkilöä, mutta liiketoimintayksiköstä haastateltiin kahta. Haastateltavina oli yksiköiden johtajien lisäksi projektipäälliköitä ja kehittäjiä, jotta tutkimuskohteesta saatiin mahdollisimman paljon tietoa. Aihetta ei oltu tutkittu aikaisemmin, joten tutkimusmenetelmäksi valittiin tämän vuoksi haastattelututkimus.

## TULOKSET

Soneran teknologiayksikön projektikehitysmalli muistuttaa hyvin paljon perinteisiä kehitysmalleja, kuten vesiputous- ja vaihe-portti-mallia. Haastattelutulosten perusteella nähtiin, että ketterät mallit on tehty juuri lieventämään tai poistamaan haasteita, joita perinteisissä malleissa ilmenee. Tämä tutkimus vahvistaa teoriaosuudessa esitettyjä väitteitä ketterien ja perinteisten menetelmien eroavaisuuksista.

Avainsanat: ketterä, tuotekehitys, palvelukehitys, lean, agile, kehitys, lean-startup, sonera, skaalaus, johtaminen

Author: Tommi Luhtala

Title: Agile and customer-centric service development in telecom industry

Date: 15.5.2016

Language: Finnish

Number of pages: 6+62

Department of Communication Networks

Professorship: Network Economics

Supervisor: Prof. Raimo Kantola

Advisor: M.Sc. (Tech.) M.Sc (Econ.) Jani Koskinen

## OBJECTIVE OF THE STUDY

In this study, the aim was to find out the benefits of different product developing methods for Sonera's service development. The study focused in all aspects of the current service development process. The objective was to find the latest and most advanced development methods and to relate them to current service development. The thesis studies how service development should be led and what kind of requirements these new development methods place leaders.

## MATERIALS

14 interviews were conducted for the study involving persons from different units. Three of the units had four interviewees, but the business unit had only two interviews. Interviewees were leaders of the unit, project managers and developers, so information could be gathered as much and broadly as possible. This subject has not been studied in the past, so interviews was chosen for the research method.

## RESULTS

Sonera's technology unit project development model is very similar to traditional development models, such as waterfall- and stage-gate-model. Based on the interviews results the agile models are made just to mitigate or eliminate the challenges, which occurs in traditional models. This study confirms the arguments in differences between agile and traditional methods presented in the theoretical part of this study.

Keywords: Agile, Lean, product development, development, service development, lean-startup, sonera, scaling, leading

## Esipuhe

Aluksi haluan kiittää valvojaani Professori Raimo Kantolaa hyvästä ohjauksesta. Haluan erityisesti kiittää ohjaajaani Jani Koskista loistavasta ohjauksesta ja valmentamisesta diplomityöni aikana. Jani Koskisen lisäksi haluan kiittää myös Soneran organisaatiota, joka tarjosi minulle hyvin mielekkään tutkimuskohteen, sekä kaikki työkalut diplomityöni saavuttamiseksi. Haluan myös kiittää kaikkia haastateltaviani, jotka suostuivat ilomielin haastatteluihin kiireidenkin keskellä, ja tarjosivat avoimesti tärkeää tietoa tutkimuskohteestani. Haluan myös osoittaa kiitollisuuttani kaikille niille henkilöille, joiden kanssa olen päässyt keskustelemaan diplomityöstäni tai siihen liittyvistä asioista.

Otaniemi, 7.4.2016

Tommi E. Luhtala

# Sisällysluettelo

<b>Tiivistelmä</b>	<b>ii</b>
<b>Tiivistelmä (englanniksi)</b>	<b>iii</b>
<b>Esipuhe</b>	<b>iv</b>
<b>Sisällysluettelo</b>	<b>v</b>
<b>1 Johdanto</b>	<b>1</b>
1.1 Taustat ja aiheen esittely . . . . .	1
1.2 Tavoitteet ja rajaukset . . . . .	1
1.3 Tutkimuskysymykset ja -menetelmät . . . . .	2
1.4 Tutkimuksen rakenne . . . . .	3
<b>2 Palvelukehitysmallit</b>	<b>4</b>
2.1 Ketterät ja asiakaslähtöiset palvelukehitysmallit . . . . .	4
2.1.1 Agile Manifesto ketterään ohjelmistokehitykseen . . . . .	4
2.1.2 XP-kehitysmalli . . . . .	6
2.1.3 Scrum-kehitysmalli . . . . .	8
2.1.4 Lean-menetelmä . . . . .	10
2.1.5 Kanban-kehitysmalli . . . . .	13
2.1.6 Lean startup-malli . . . . .	13
2.2 Perinteiset tuotekehitysmallit . . . . .	17
2.2.1 Vaihe-portti-malli . . . . .	17
2.2.2 Vesiputousmalli . . . . .	18
2.3 Perinteisten tuotekehitysmallien ero ketteriin . . . . .	20
2.3.1 Kehitysmallien erot teoreettisella tasolla . . . . .	20
2.3.2 Kehitysmallien suorituserot projekteissa . . . . .	21
2.4 Ketterän palvelukehitysmallin skaalaaminen isossa yrityksessä . . . . .	24
2.4.1 SAFe-kehitysmalli . . . . .	24
2.4.2 MAXOS-kehitysmalli . . . . .	26
2.5 Ketterän palvelukehityksen johtaminen . . . . .	27
2.5.1 Ketterän johtamisen onnistuvuus . . . . .	28
2.5.2 Käytännön ohjeita ketterään johtamiseen . . . . .	31
<b>3 Tutkimusaineisto ja -menetelmät</b>	<b>32</b>
3.1 Tutkimusmenetelmä ja sen perustelu . . . . .	32
3.2 Tutkimushaastatteluiden toteutus . . . . .	32
3.3 Haastatteluaineiston analysointi . . . . .	34
<b>4 Tulokset</b>	<b>36</b>
4.1 Haastatteluiden tulokset . . . . .	36
4.1.1 Ongelmat palvelukehityksessä . . . . .	36
4.1.2 Palvelukehityksen kehittäminen ja henkilöresurssit . . . . .	37

4.1.3	Mittaaminen ja priorisointi . . . . .	38
4.1.4	Palvelukehityksen johtaminen ja projektien lopettaminen . . .	40
4.1.5	Ketterien ja perinteisten menetelmien erot ja asiakkaan rooli .	41
4.2	Soneran palvelukehitysprosessi . . . . .	43
<b>5</b>	<b>Johtopäätökset ja analyysi</b>	<b>47</b>
5.1	Tulosten analysointi . . . . .	47
5.1.1	Projektikehitysmalli . . . . .	47
5.1.2	Hankkeiden laajuus ja mittaaminen . . . . .	48
5.1.3	Yleisimmät haasteet palvelukehityksessä ja tiimien johtaminen	49
5.1.4	Ketterien menetelmien tuomat muutokset ja asiakkaan rooli .	50
5.1.5	Hankkeiden lopettaminen ja priorisointi . . . . .	52
5.2	Kehitysmallien hyödyt Soneran palvelukehityksessä . . . . .	53
5.3	Tulevaisuuden tutkimuskohteet . . . . .	56
<b>6</b>	<b>Yhteenveto</b>	<b>57</b>
	<b>Viitteet</b>	<b>59</b>

# 1 Johdanto

Tässä luvussa esitellään tutkimusta ja tuodaan ilmi syitä miksi tämä aihe on tärkeä ja vaatii tutkimista. Tutkimuksen taustalla olevien syiden lisäksi käydään läpi tämän työn tavoitteita ja motivaatiota. Tutkimuksen rajauksesta ja valituista näkökulmista kerrotaan tässä luvussa lisää ja tuodaan ilmi seikkoja, jotka vaikuttavat tutkimuskoh- teeseen, mutta on päätetty jättää työn ulkopuolelle. Lisäksi tässä luvussa käydään läpi tutkimusmenetelmät ja esitellään tutkimuskysymykset. Viimeisessä alaluvussa on kuvattu tutkimuksen rakenne.

## 1.1 Taustat ja aiheen esittely

Menestyksestä liiketoimintaa harjoittavan yrityksen on tarjottava hyviä palveluita tai tuotteita. Viimeisten vuosien aikana asiakkaan rooli palvelukehityksessä on nous- sut entistä tärkeämmäksi. Kansainvälistymisen myötä yhä useampi yritys tarjoaa palveluita, joiden tehtävänä on täyttää asiakkaiden muuttuvia tarpeita. Myös entistä nopeammin kehittyvä teknologia mahdollistaa monipuolisempia ja tyydyttävämpiä ratkaisuja asiakkaille. Tämän vuoksi palvelukehityksen haasteena on tuottaa entistä nopeammin entistä monipuolisempia ratkaisuja, jotka kuitenkin vähintäänkin täyttä- vät ja mieluummin ylittävät asiakkaan odotukset. Samalla, kun asiakkaiden tarpeet ovat muuttuneet haastavammiksi, niin usein tuotekehityskuluja haluttaisiin vähentää tai vähintäänkin säilyttää ennallaan.[1][2]

Palvelukehitystä varten on olemassa erilaisia menetelmiä ja kehitysmalleja, joiden tulisi auttaa kehittämään parempia palveluita. Suuret yritykset ovat ottaneet oppia menestyneiltä Startup-yrityksiltä ja hyviä menetelmiä on pyritty tuomaan kaikkiin yrityksiin. Ketterien ja Lean-menetelmän malleja on sovellettu asiakaslähtöiseen palvelumuotoiluun ja yhdistetty näitä palvelukehitykseen. Tutkimuksen tarkoituk- sena on selvittää keinoja palvelukehitysprosessin tuottavuuden lisäämiseen uusien palvelukehitysmallien avulla. Diplomityö tehdään Soneralle, jossa ollaan siirtymässä vanhoista kehitysmalleista uudempiin tuotekehitysmalleihin.

## 1.2 Tavoitteet ja rajaukset

Tässä tutkimuksessa keskitytään eri palvelukehitysmallien hyötyihin ja eroavaisuuksiin. Palvelukehitysprosessien kehittämisestä on tehty aiempia tutkimuksia. Eräässä- kin tutkimuksessa on keskitytty tarkemmin organisaatiossa tehtäviin muutoksiin[3]. Muutos palvelukehitysprosessista täysin erilaiseen malliin on isossa yrityksessä var- masti suuri haaste, mutta tämän työn tarkoituksena ei ole paneutua muutokseen vaan mallien hyötyihin. Palvelukehitysprosessiin vaikuttaa moni asia ja onnistuneen palvelun kehittäminen voi tapahtua vasta jatkokehityksen aikana. Tämän vuoksi palvelukehityksen tehokkuutta on hankala mitata, kun liiketoiminnallisesti tuotta- van palvelun kehitys ei riipu yksinomaan kehitysprosessista. Tässä työssä ei käydä

tuotantoon siirtämisen osa-alueita läpi, vaikka se on hyvin tärkeä osatekijä palvelun tuottavuuden ja kannattavuuden kannalta. Tässä työssä ei myöskään tutkita yrityksen arvojen vaikutusta palvelukehitykseen, vaikka niilläkin on oma osuus prosessin tehokkuudessa. Käytännön tutkimusosuus rajoittuu sekä kilpailullisten rajoitteiden, että aikarajoitteiden vuoksi ainoastaan Soneran palvelukehitykseen.

Työn pääasiallisena tavoitteena on löytää selvät hyödyt erilaisille palvelukehityksmallille. Näiden hyötyjen tuominen yritysten tietoon voi auttaa yrityksiä parantamaan omaa palvelukehitystään ja innovointiaan. Työ keskittyy palvelukehitykseen prosessina sekä osana organisaatiota, jotka ovat pääosin monistettavissa esimerkiksi teknologiateollisuuden yrityksiin. Tavoitteena on myös löytää toimeksiantona tehtävälle teleliiketoiminnan yritykselle optimaalisin palvelukehitysmalli.

### 1.3 Tutkimuskysymykset ja -menetelmät

Tutkimuskysymykset liittyvät palvelukehitysprosessin ja palvelukehityksen kehittämiseen ja johtamiseen. Tavoitteena on tutkia mitä hyötyä uusista ketteristä ja asiakaslähtöisistä tuote- ja palvelukehityksmallista on suuren yrityksen palvelukehitysprosessille. Nämä mallit eroavat käytännöllisten menetelmien lisäksi myös ajatusmaailmaltaan paljon toisistaan, joten muutos koskee myös vahvasti johtamista. Tämän vuoksi on tärkeää ottaa kantaa työssä itse prosessien lisäksi myös palvelukehityksen johtamiseen. Palvelukehitysprosessin tehokkuutta voidaan mitata läpimenoajoilla, tuottavuudella, asiakkaiden tyytyväisyydellä sekä omien työntekijöiden tyytyväisyydellä. Tutkimuskysymykset liittyvät edellä mainittuihin aiheisiin ja tutkimuksen pääkysymys on esitetty ensimmäiseksi:

- Miten palvelukehitysprosessia voidaan tehostaa ketterien tuotekehityksmallien avulla?
- Miten ketterää palvelukehitystä tulee johtaa?

Tutkimusaineistona käytetään haastattelututkimusta, sekä havainnointia palvelukehitysprojektissa. Tällä tavoin pyritään saamaan mahdollisimman hyvä kokonaiskuva ennalta tutkimattoman palvelukehitysprosessin eri osa-alueista. Haastatteluilla pyritään löytämään prosessin eri osa-alueiden toimintaa, sekä kuvaamaan mitä kehitettäviä asioita nykyisessä prosessissa on. Haastatteluilla pyritään kuvaamaan myös nykyisen organisaation valmiuksia siirtyä tukemaan ketteriä ja asiakaslähtöisiä palvelukehityksmallia. Haastateltavia ihmisiä haetaan eri puolelta organisaatiota, jotta tietoa saadaan kerättyä palvelukehitysprojeekteista mahdollisimman laajasti. Tietoa haetaan palvelukehityksen johtamisesta, rahoittamisesta, ideoinnista ja projektien toteutuksesta. Haastattelujen tukena hyödynnetään havainnointia käynnissä olevista palvelukehitysprojeekteista.



## 1.4 Tutkimuksen rakenne

Johdannon jälkeen siirrytään teoriaosuuteen, joka käsittelee ketterien ja asiakaslähtöisten palvelukehitysmallien lisäksi perinteisiä tuotekehitysmalleja. Työssä esitetään viimeisin tieto uusimmista sekä käytetyimmistä tuote- ja palvelukehitysmalleista. Tässä osuudessa on tärkeätä tuoda viimeisin akateeminen tieto, sekä jo yrityksistä saatu tieto ja yhdistää näitä maailmoja. On hyvä huomata, että työelämässä ja varsinkin suurissa yrityksissä toimintamallit voivat olla vuosikymmenten takaa. Toisaalta luotettavaa tietoa ei ole saatavilla uusimpien palvelukehitysmallien osalta. Toisessa luvussa käsitellään myös näiden mallien eroja ja hyötyjä. Käsiteltäviä asioita ovat operatiivinen toiminta, johtaminen, arvot ja rahoitus.

Kolmannessa luvussa siirrytään tutkimusaineiston ja -menetelmien pariin, jossa määritellään, miten palvelukehityksen kehittämistä voidaan tehdä. Luvussa käydään läpi mittaristo, jolla palvelukehityksen hyvyttä voidaan mallintaa. Luvussa käydään läpi millä menetelmillä tietoa haetaan valmistuneista projekteista sekä käynnissä olevista projekteista. Luvussa esitetään, miten haastattelut suoritetaan ja millä menetelmällä tietoa kerätään käynnissä olevasta palvelukehitysprojektista, jossa kirjoittaja on ollut mukana ideoimassa ja suunnittelee palvelun prosesseja.

Neljäs luku esittelee haastatteluiden sekä valmistuneiden palvelukehitysprojektien avulla saatuja tuloksia. Luvussa esitetään tulosten avulla palvelukehityksen nykytila, jonka jälkeen esitellään erikseen haastatteluiden tulokset ja projekteista saatujen tietojen tulokset. Luvun lopussa yhdistetään tulokset ja analysoidaan näitä.

Viidennessä luvussa esitetään johtopäätökset, kehitysehdotukset ja yhteenveto. Johtopäätökset tehdään yhdistämällä teoriaosuus käytännön tuloksiin. Näiden analysointi antaa valistuneempaa tietoa siihen mikä on yrityksen nykytila, sekä mikä olisi hyvä suunta tulevaisuuteen. Tämän jälkeen esitetään kehitysehdotukset yritykselle ja palvelukehitysprosessiin. Kehitysehdotukset esitellään myös erikseen, miten palvelukehitystä tulisi johtaa. Loppuun esitetään vielä yhteenveto, jossa käsitellään, onko tutkimuskysymyksiin vastattu hyvin ja mitä hyötyä tutkimuksesta on ollut. Luvussa käsitellään myös tulevaisuutta ja miten palvelu- tai tuotekehitystä tulisi tutkia jatkossa.

## 2 Palvelukehitysmallit

Tässä luvussa käydään läpi eri palvelukehitysmalleja läpi teoreettisella tasolla. Kehitysmallien eroja käydään läpi teorian lisäksi myös käytännön onnistuvuuden perusteella. Luvun lopussa tarkastellaan vielä ketterän palvelukehityksen johtamista ja siihen liittyviä odotuksia.

### 2.1 Ketterät ja asiakaslähtöiset palvelukehitysmallit

Ketterät kehitysmallit on alun perin suunniteltu ohjelmistokehitykseen ja ne alkoivat yleistyä 90-luvun puolivälissä. Ketterämpien mallien oli tarkoitus tuottaa ohjelmistoa nopeammin ja käyttää resursseja tehokkaammin[1]. Ketteriä kehitysmenetelmiä on useita erilaisia, eikä yhtä oikeaa tai universaalista kehitystapaa ole olemassa[2]. The Agile Manifesto oli ensimmäinen julkaisu, joka käsitteli ketteriä kehitysmenetelmiä perinpohjaisesti.

#### 2.1.1 Agile Manifesto ketterään ohjelmistokehitykseen

Suurin osa ketterän kehityksen malleista perustuu 2001 tehtyyn Agile Manifesto-julkaisuun. Julkaisun tarkoitus oli parantaa ohjelmistokehitysprosessia ja tehdä siitä tuottavampaa[3]. Kuvassa 1 näkyy hyvin, että ketterät menetelmät keskittyvät yhteistyön etuihin. Monet muutoksista pyrkivät siihen, että asioista keskustellaan ja sovitaan yhdessä. Lopputulosta pidetään tärkeämpänä kuin suunnitelmaa, jolla siihen on päästy.

Sen ensimmäinen sanoma on, että yksilöt ja heidän vuorovaikutus tulisi asettaa prosessien ja järjestelmien edelle. Parhaatkaan prosessit ja työkalut eivät tee kehityksestä tehokasta, jos ihmiset eivät työskentele hyvin yhdessä. Ohjelmistokehityksessä tiimin tulisi sisältää ohjelmoijia, testaaajia, projektijohtajia ja asiakkaita. Näiden henkilöiden keskinäinen yhteistyö on avainroolissa ketterässä ohjelmistokehityksessä.[4]

Julkaisun toisen arvon mukaan toimiva ohjelmisto on tärkeämpää kuin tarkka dokumentointi. Tämä ei tarkoita, että dokumentointia tulisi unohtaa, koska siitä ilmenee miksi jokin järjestelmä tehdään ja mitä järjestelmiä ohjelmisto hyödyntää. Pääpaino kehityksessä on kuitenkin toimivan ohjelmiston tuottamisessa asiakasta varten. Suurin osa asiakkaistakin arvostaa toimivaa ohjelmistoa enemmän kuin hyviä dokumentteja tai kaavioita.[4]

Kolmas arvo julkaisussa sanoo, että asiakasyhteistyö on tärkeämpää kuin sopimusneuvottelut. Asiakkaan tarpeen tyydyttämiseksi on huomattavasti helpompi tehdä yhteistyötä asiakkaan kanssa kuin arvailla mitä pitäisi tehdä. Asiakkaalla ei usein ole täyttä ymmärrystä, miten ohjelmisto rakennetaan, joten alkuperäisiin suunnitelmiin tulee todennäköisesti muutoksia. Vastuiden ja oikeuksien sopiminen helpottaa yhteistyötä ja luo pohjan sopimukselle, mutta tämä ei korvaa kommunikoinnin tarvetta.

Onnistuneissa kehitystöissä tehdään usein paljon yhteistyötä asiakkaan kanssa, jotta asiakkaan muuttuvat tarpeet täytetään jatkuvasti.[4]

Viimeisenä arvona julkaisu kehottaa vastaamaan muutokseen mieluummin kuin pysymään alkuperäisessä suunnitelmassa. Muutoksen tekeminen on tärkeää, koska ihmiset muuttuvat, liiketoimintakenttä muuttuu ja teknologiat voivat muuttua. Kesken kehityksen opitaan jatkuvasti lisää, joka voi myös muuttaa alkuperäisiä suunnitelmia, kun tehdään jokin uusi löydös. Muutoksista huolimatta projektisuunnitelmaa ei saa unohtaa, vaan ideana on reagoida muutoksiin ja muuttaa alkuperäistä suunnitelmaa, kun muutostarpeita ilmenee.

Yksi tärkeimpiä ominaisuuksia ketterässä kehittämisessä onkin nopea reagoiminen ja alkuperäisten suunnitelmien muuttaminen.[1] Ketterä kehitys vaatii tehokasta kommunikointia myös tiimien sisällä, jotta ryhmä pystyy vastaamaan muutoksiin mahdollisimman nopeasti. Ketteryys vaatii myös organisaatiolta sitä, että tiimillä on täysi hallintaoikeus omasta toiminnastaan. Tällöin tiimi voi tehdä itse päätökset nopeasti eikä sen tarvitse järjestää palavereja päättäjien kanssa. Näiden nopeaa toimintaa mahdollistavien tekijöiden lisäksi on tärkeää, että asiakas tuodaan mahdollisimman lähelle tiimiä, jotta kehitys on läpinäkyvää ja asiakkaan toiveet voidaan toteuttaa ilman viiveitä.[3]

## Agile manifesto: Perusperiaatteet

Yksilöitä & Kanssakäymistä	Enemmän kuin	Menetelmiä & Työkaluja
Toimivaa ohjelmistoa	Enemmän kuin	Kattavaa dokumentointia
Asiakasyhteistyötä	Enemmän kuin	Sopimusneuvotteluja
Vastaamista muutokseen	Enemmän kuin	Pitäytymistä suunnitelmassa

Kuva 1: Agile manifesto-julkaisun perusperiaatteet

Ketterät kehitysprosessit omaavat myös lyhyen aikavälin suunnitelmat, joita tarkastellaan periodeissa ja suunnitelmat pohjautuvat lopulta aina asiakkaan tarpeisiin. Muutoksia on paljon helpompi tehdä kun suunnitelmat ovat vain lyhyelle aikavälille

jolloin turhaa työtä ei kerry. Ketterän kehityksen tärkein periaate on tyydyttää asiakastarpeet jatkuvalla toimitussyklillä. Muita periaatteita ovat tehokas muutoksenhallinta, jotta muutosten tekeminen ei olisi kallista kehityksen loppupuolellakaan. Ketterässä kehityksessä liiketoiminnasta vastaavien henkilöiden ja kehittäjien tulisi työskennellä yhdessä päivittäin koko projektin ajan. Lyhyen aikavälin suunnitelmien lisäksi on tärkeää, että on olemassa, myös pitkän aikavälin suunnitelma jota vasten kokonaisuuksia pilkotaan pienemmiksi tehtäviksi.[4]

Ensimmäinen laajalti käyttöön otettu ketterä kehitysprosessi oli nimeltään XP (engl. Extreme Programming), joka rakentuu käyttäjäkertomusten (engl. User Stories) ympärille[3]. Käyttäjäkertomuksia kerätään yhteen ja näiden pohjalta suoritetaan tehtäviä, jotta saadaan toimitettua asiakkaalle hänen tarvitsemansa ratkaisu.[4][1][3]

### 2.1.2 XP-kehitysmalli

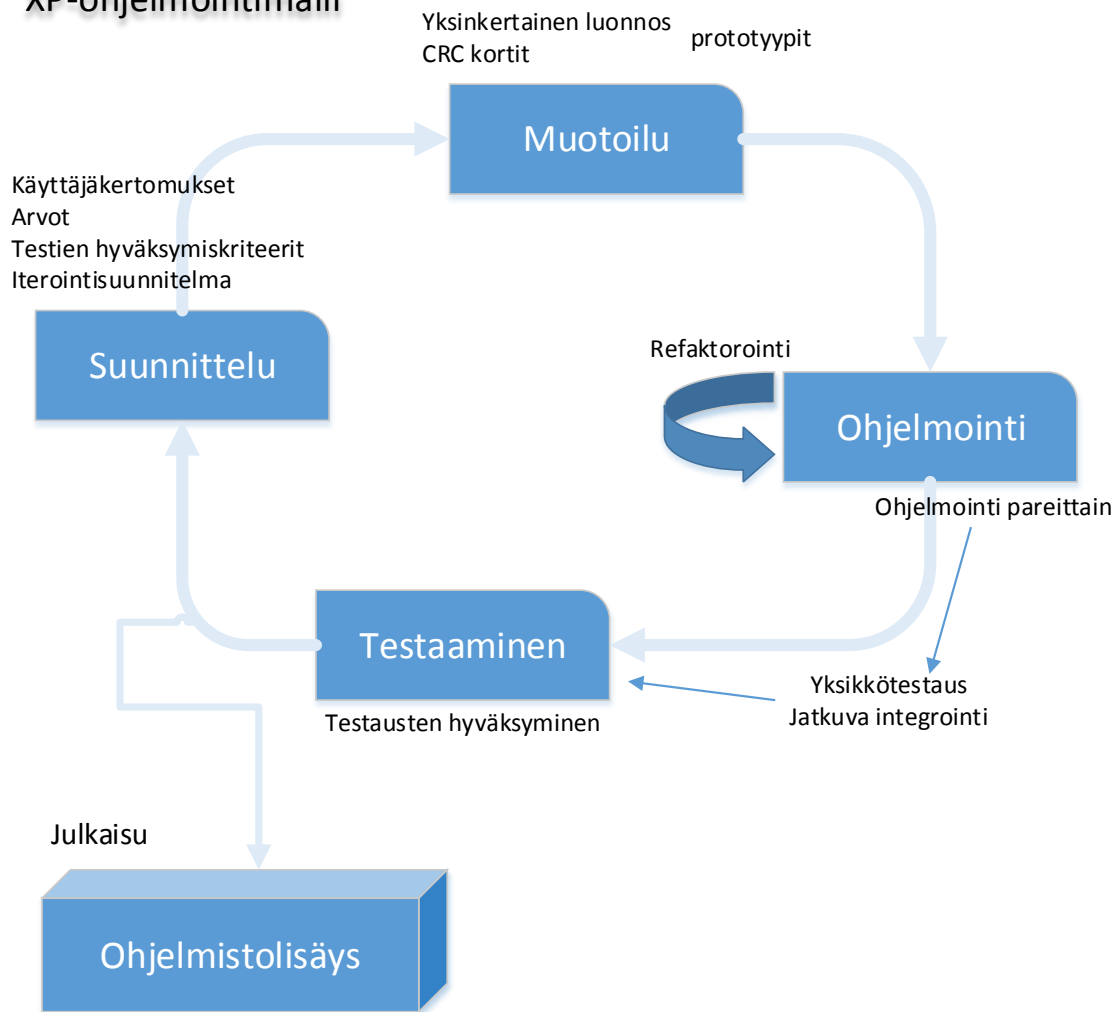
XP-menetelmä kehitettiin pieniä tiimejä varten, jotka työskentelevät jatkuvasti muuttuvassa ja epävarmassa ympäristössä. Kuvassa 2 näkyy tiimin tehtävät, jotka riittävät valmiin ohjelmisto-osan tuottamiseen. XP-malli rakentuu neljän perusarvon ympärille, jotka ovat samansuuntaisia kuin ketterässä ohjelmistokehityksessä:

1. *Kommunikointi*: Suuri osa epäonnistuneista projekteista johtuu huonosta kommunikoinnista. Kommunikointi on tiimin tärkein arvo ja se on työskentelyn perusta. Kommunikointi käsittää ryhmän sisäisen tiedon vaihtamisen lisäksi myös asiakkaiden, ohjelmoijien ja esimiesten kanssa käytävän kommunikoinnin.
2. *Yksinkertaisuus*: On tärkeää lähteä liikkeelle yksinkertaisimmasta mahdollisesta ratkaisusta joka toimii. Ohjelmoijien tulisi ratkaista nykypäivän ongelma nykyisillä työkaluilla ja antaa tulevien ratkaisuiden tai teknologioiden pysyä tulevaisuudessa. Monimutkaisten ja mullistavien ratkaisuiden kehittäminen maksaa usein liikaa eikä niistä saatava hyöty korvaa menetettyä aikaa ja rahaa. [5]
3. *Palaute*: XP-mallin käytännöt on suunniteltu hyödyntämään palautetta heti alusta alkaen ja tiheään tahtiin. Testaus antaa luotettavaa ja ajantasaista palautetta, kun työtä tehdään lyhyissä aikajaksoissa tai jatkuvana kehityksenä. Jatkuva palaute, ja sen hyödyntäminen ohjaa projektia koko ajan oikeaan suuntaan.
4. *Rohkeus*: Monet kehitysmallit ovat ympäripyöreitä ja keskittyvät kaiken varmistamiseen ja tekemiseen, joka on oikeissa projekteissa vaikea toteuttaa. XP-malli korostaa tekijöitään ottamaan riskin voittaakseen ja siirtyä käytännön tekemiseen monia malleja nopeammin.

[6]

XP-malli koostuu 12 käytänteestä, joita ohjaavat edellä mainitut neljä perusarvoa.

## XP-ohjelmointimalli



Kuva 2: XP-ohjelmointimalli yksinkertaistettuna

*Suunnittelu:* Tarkoitus on määrittää työn laajuus yhdessä asiakkaan kanssa, jolloin saadaan kerralla liiketoiminnalliset tavoitteet ja tekniset vaatimukset projektin tekemiseksi.

*Pienet julkaisut:* Saadaan systeemi ja sen liitännäiset nopeasti tuotantoon. Nopeat julkaisut ovat tärkeä tekijä palautteen saamiseksi ja projektin ohjaamista varten.

*Metafora:* On tärkeää ymmärtää, miten koko järjestelmä toimii. Asiakkaan ja ohjelmoijien täytyy kaikkien ymmärtää, miten kokonaisuus toimii ja miten sen eri osat liittyvät toisiinsa.

*Yksinkertainen malli:* Yksi perusarvoista XP-mallissa on yksinkertaisuus. Järjestelmään pitäisi suunnitella vain sellaisia osia, joita aiotaan liittää siihen lyhyellä aikavälillä. Tulevaisuuden ennustaminen ja sen varaan komponenttien suunnitteleminen on hankalaa ja menee usein pieleen.

*Testaus:* Palaute on myös yksi perusarvoista, joten se on ensiarvoisen tärkeää XP-mallissa. Testeihin sisältyvät useat erilaiset kokeet ja ne kertovat projektin valmiusasteen.

*Refaktorointi:* Refaktorointi on prosessi, jossa koodia parannellaan ilman, että sen toiminnallisuus muuttuu. Tämän tarkoituksena on parantaa luettavuutta ja tämä on ohjelmoijien päivittäistä tekemistä.

*Pareittain ohjelmointi:* Työskentely XP-mallissa tapahtuu pareittain kun ohjelmoidaan koodia tuotantoa varten.

*Yhteinen omistus:* Kuka tahansa tiimistä voi muuttaa mitä tahansa osaa järjestelmästä. Tiimi on siis tasa-arvoinen eikä kukaan ole vastuussa vain tietystä osasta järjestelmää.

*Jatkuva ohjelmointi:* Ohjelmoijat muuttavat ja kehittävät järjestelmää jatkuvasti ilman ennalta määrättyä julkaisusykliä.

*40 tuntia viikossa:* XP-mallissa tulisi tehdä töitä 40 tuntia viikossa vaikka usein ohjelmistokehityksessä se tarkoittaa töiden tekemistä niin pitkää kuin jaksaa.

*Asiakas läsnä:* Asiakas kuuluu tiimiin ja hänen tulisi olla kokopäiväisesti paikalla vastaamassa kysymyksiin ja auttamassa kehittäjiä. Asiakkaan vastuulla on myös määritellä hyväksyttämistestit.

*Standardoitu koodi:* Kommunikointi oli tärkein perusarvo XP-mallissa ja se heijastuu myös kirjoitettavaan koodiin. Kollektiivisessä omistuksessa oleva järjestelmä tulisi sisältää standardoitua koodia, joita kaikki ymmärtävät, jotta kommunikointi helpottuu. [7]

### 2.1.3 Scrum-kehitysmalli

Scrum on ketterä menetelmä innovatiivisten tuotteiden ja palveluiden kehittämiseen[8]. Scrum-mallin käyttäminen ja käyttöönotto vaatii organisaatiolta jatkuvaa kokeilua ja oppimista[7]. Scrum-mallin pienin osa on tiimi, joka koostuu usein 3-10 henkilöstä. Tiimillä on oikeus päättää itse miten ratkaisu toteutetaan, mutta tiimille annetaan ylempää ongelma tai tehtävä, joka tulee ratkaista. Mallissa on myös ominaista ajalliset jaksot, jotka ovat usein kahdesta kuuteen viikkoon. Jokaisen jakson aikana tulee saada tietty tehtävä tai useampia tehtäviä valmiiksi. Jokaisen jakson jälkeen pidetään katsaus, että mitä ollaan saavutettu ja millä tavoin. Tämän katsauksen tarkoitus on sekä kehittää työntekoa, että organisaatiota toimimaan paremmin Scrum-menetelmän avulla. On tärkeää muistaa, että ei ole yhtä oikeaa tapaa toteuttaa Scrum-mallia. Käyttö vaatii jatkuvaa kokeilua ja opettelua, jotta siitä saataisiin täysi hyöty.[8]

Scrum-kehitysmalli on alun perin suunniteltu ohjelmistokehitystä varten, johon se myös sopii parhaiten. Scrum-menetelmien positiivisten tulosten vuoksi mallia on myös ruvettu hyödyntämään muilla aloilla ja varsinkin tuote- ja palvelukehitystä vaativissa töissä. Scrum-malli kiteytyy erilaisiin tehtävälistoihin, jotka voidaan jakaa

isommassa mittakaavassa tuotteen tai palvelun tehtävälistaan. Tämä tehtävälista jaetaan vielä tiimien aikajaksoihin sopiviksi pienemmiksi tehtävälistoiksi, joihin tiimi pureutuu ja joihin se koittaa hakea parhaita mahdollista toteutusta. Tiimi koostuu omistajasta, johtajasta ja kehittäjistä. Mallin tehokkuus perustuukin siihen, että tiimi on itseohjautuva ja se pystyy päättämään itse käyttämistään menetelmistä ja miten tehtävälistan toteutus hoidetaan.[8]

Scrum-tiimin tuoteomistaja (engl. Product Owner) vastaa tiimin liiketoiminnallisesta tuloksesta. Tuoteomistaja vastaa käyttäjistä ja asiakkaista sekä huolehtii siitä, että kehitettävä tuote tuottaa mahdollisimman paljon arvoa sekä yritykselle, että asiakkaille. Omistaja huolehtii siitä, että tehtävälistaan valitut tehtävät ovat liiketoiminnan kannalta tärkeitä ja kasvattavat tuotteen arvoa eniten. Omistaja on myös mukana määrittelemässä vaatimuksia ja tarkentaa niitä tiimille, mikäli tarvetta määrittelylle ilmenee.[7]

Scrum-tiimin johtaja (engl. Scrum Master) vastaa tiimin työmenetelmistä ja päivittäisestä tekemisestä ja pitää huolen siitä, että tiimi työskentelee Scrum-metodien mukaisesti. Scrum-johtaja ei ole tiimin jäsenten esimies vaan johtaminen tapahtuu Scrum-prosessien kautta. Johtajan tehtävä tiimissä on huolehtia prosesseista ja poistaa häiriötekijöitä, jotta tiimi suoriutuu tehtävistään mahdollisimman tehokkaasti ja laadukkaasti. Johtajan tärkein tehtävä on pitää tiimin tehtävälista sopivan kokoisena ja huolehtia siitä, että aikajakson tai pyrhdyksen aikana tiimi saa tehtävänsä valmiiksi.[7]

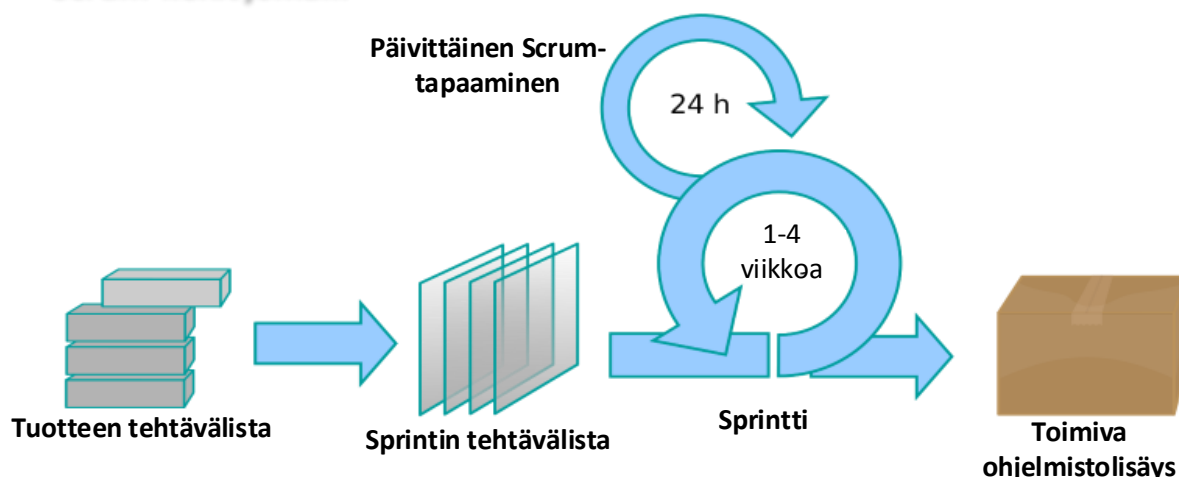
Tuoteomistajan ja johtajan lisäksi Scrum-tiimi koostuu kehittäjistä (engl. Development Team) jotka ovat kehitettävän tuotteen tai palvelun asiantuntijoita tai muuten osaavat edesauttaa tehtävien suorittamisessa. Ohjelmistoprojekteissa tiimissä on useita ohjelmoijia, mutta palvelukehityksessä tiimi koostuu usein eri osa-alueiden asiantuntijoista.[7]

Scrum-kehitysmalli on hyvin järjestelmällinen ja menetelmän käyttöönotto vaatiikin usein vahvaa sitoutumista Scrum-menetelmän prosesseihin ja käytäntöihin. Vaikka menetelmät ovatkin ehdottomia ja välillä jopa raskaitakin, niin niillä on tärkeä tarkoitus ja ne edesauttavat tiimin suoriutumista eivätkä estä luovaa ja ketterää työskentelyä. Scrum-kehitysmalli selkeä, jossa tavoitteita pilkotaan suurista kokonaisuuksista pieniin osatekijöihin. Kuvassa 3 on koko kehitysmalli yksinkertaistettuna.

Pyrähdys (engl. Sprint) on aikajakso, jonka aikana tiimin on saatava valmiiksi sille asetetut tehtävät. Pyrhdyksen pituus on vähintään yhden viikon ja enintään kuukauden mittainen. Myös tehtävän valmiusaste on määriteltävä tarkkaan koko ohjelman kannalta, jotta tiimit tietävät mihin heidän on pyrittävä, jotta pyrhdyksen aikana tehdyt tehtävät ovat kaikkien mielestä valmiita.[7][8]

Pyrähdys koostuu aloituspalaverista, jossa suunnitellaan tehtävälista ja jaetaan työtehtäviä kategorioittain sekä hahmotellaan pyrhdyksen aikana tehtävää työtä. Päivittäin on pidettävä myös palaveri, joka usein pidetään aamulla, jolloin kehitystiimi suunnittelee päivän aikana tehtävät asiat sekä tasaa työkuorman ja synkronoi toimintansa. Tämä on hyvin tärkeää varsinkin, jos tiimi on hajautunut useampaan

## Scrum-kehitysmalli



Kuva 3: Scrum-kehitysmallin metodologia

paikkaan ja varsinkin jos kehittäjät työskentelevät eri puolella maapalloa. Palaverissa on tarkoitus selvittää, mitä työtä ollaan tehty, mitä ollaan tekemässä ja mitä pitäisi saada aikaan ennen seuraavaa päiväpalaveria (engl. Daily Scrum). Päiväpalaverin tarkoitus on olla lyhyt alle 30 minuutin nopea kokoontuminen, jossa käydään asiat nopeasti läpi, tiedetään, mitä kukakin on tekemässä ja mistä päivä koostuu. Pyrähdyksen päätyttyä pidetään loppukatselmus (engl. Sprint Review), jossa tarkastetaan tehtävät ja julkaistu materiaali. Tässä kohtaa arvioidaan pyrähdyksen aikana tehdyn tuotoksen laatu ja usein pidetään esittelytilaisuus, jossa näytetään tiimin aikaansaannokset. Tästä tilaisuudesta tiimi saa palautetta demonsa perusteella, jonka jälkeen se voi määritellä lisääkö seuraavan pyrähdyksen tehtävälistaan uusia tehtäviä ja miten vai mennäänkö alkuperäisen suunnitelman mukaan. Katselmuksen tarkoituksena on luoda hyvä pohja seuraavalle suunnittelupalaverille, jossa käydään läpi seuraavan pyrähdyksen tehtävälistaa ja työjonoa läpi. Pyrähdyksessä pidetään lopuksi myös retrospektiivi, jossa käydään läpi tiimin työskentelytapoja ja -menetelmiä sekä tehdään mahdollisia korjaustoimenpiteitä. Tämä on tärkeä osa tiimin alkutaivalta, kun yritetään löytää parhaita toimintatapoja ja käytäntöjä tiimille. Myös organisaatiolle on hyötyä tästä, sillä Scrum-menetelmään siirtymisen jälkeen organisaatiolla on vielä paljon opittavaa.[7][8]

### 2.1.4 Lean-menetelmä

Lean-menetelmä voidaan kuvata filosofiana, jonka pyrkimys on toiminnan jatkuva parantaminen[9]. Filosofia kiteytyy hukkatointojen (engl. waste) kitkemiseen ja tämmöiseksi luokitellaan kaikki toiminnot, jotka eivät tuota asiakkaalle lisäarvoa[10]. Lean-ajattelu on lähtöisin Toyotalta, jossa valmistusmenetelmiä kehitettiin niin, että kustannukset laskivat, mutta laatu ja tuottavuus paranivat. Lean-ajattelun huomattiin parantavan tuottavuutta myös muissa yrityksissä, kun menetit tuotiin



länsimaihin.

Lean-ajattelua kuvataan usein talona, jonka keskiössä on ihmiset. Kuvassa 4 näkyy tämä talo ja Lean-ajattelun perustukset. Womack ja Jones on tiivistänyt Lean-ajattelun viiteen peruseriaatteeseen:[10]



Kuva 4: Lean-menetelmän perusta

1. *Määritä arvo:* Lean-ajattelu alkaa tuotteen arvon määrittämisestä, jotta voidaan tuottaa ainoastaan sitä mistä asiakas on valmis maksamaan. Kaikki työ joka ei tuota lisäarvoa asiakkaalle on turhaa ja pitää jättää tekemättä.
2. *Tunnista arvoketju:* Koko arvoketjun tunnistaminen tilausprosessista toimittukseen on Lean-ajattelun toinen askel. Koko arvoketjua tulisi tarkastella ja optimoida, jotta turhat vaiheet voidaan poistaa tai muokata tuottamaan nopeammin tai enemmän arvoa asiakkaalle.
3. *Virtaus:* Kolmas askel Lean-menetelmissä on määritellä uudelleen osastot ja tehtävät, jotta saataisiin aikaiseksi toiminnan jatkuva virtaus.

4. *Imu*: Tuotetta pitäisi valmistaa vain tarpeeseen. Juuri oikeaan tarpeeseen-menettelyn mukaisella tuotannolla vähennetään turhien varastojen määrää ja varmistetaan, että asiakas saa juuri sen mitä haluaa.
5. *Täydellisyyden tavoittelu*: Lean-ajatteluun kuuluu jatkuva toiminnan parantaminen ja kehittäminen. Käytännössä juuri neljää edellä mainittua asiaa voidaan parantaa jatkuvasti, joka tuottaa lisää arvoa asiakkaalle ja optimoi tuotantoa.[10][9]

Edellä mainittujen viiden peruseriaatteen lisäksi yksi tärkeä periaate Lean-ajattelussa on ihmisten kunnioitus. Sidosryhmät, asiakkaat, toimittajat, investoijat ja kehittäjät ovat tärkeitä resursseja joihin liiketoiminnan harjoittaminen kulminoituu. Kunnioituksen puute näitä ihmisiä kohtaan ei ainakaan luo lisäarvoa, vaan aiheuttaa hukkaa.[11] Lean-ajattelu ei päde pelkästään massatuotantoon ja tuotantoympäristöön, vaan ajattelumallia on hyödynnetty palvelualoilla ja tuotekehityksessä. Lean-menetelmistä on myös kehitetty useita eri toimintatapoja eri toimialoille ja useat yritykset hyödyntävät oppeja toiminnassaan.

Periaatteiden lisäksi Lean-menetelmät hyödyntävät useita käytännöllisiä työkaluja toiminnan edistämiseksi. Näitä käytännön työkaluja on esitelty taulukossa 1:

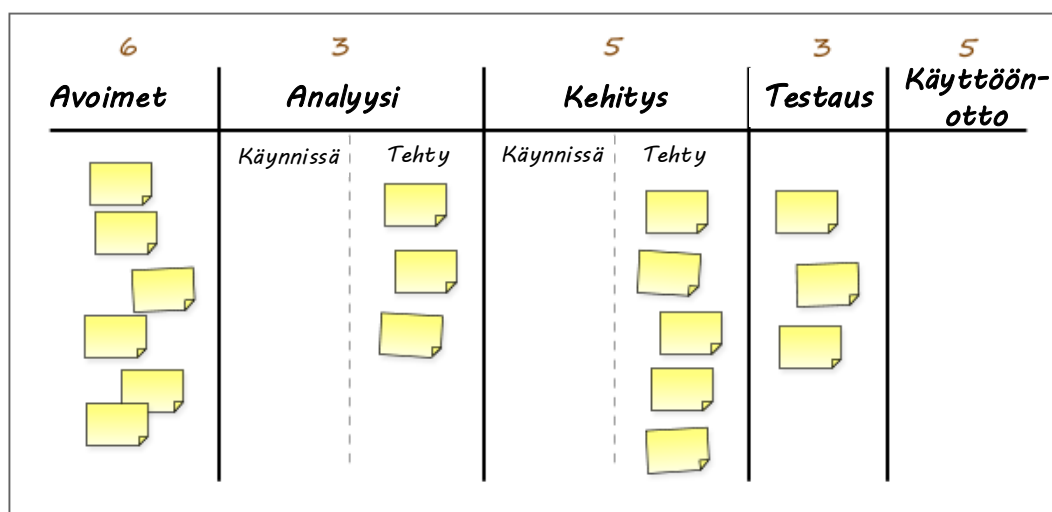
Taulukko 1: Lean-menetelmän työkaluja.

Lean-työkalu tai -prosessi	Selitys
5S	Sortteeraus, Systematisointi, Siivous, Standardisointi, Seuranta. Parantaa tuottavuutta pitämällä työympäristön järjestyksessä.
Juuri-oikeaan-tarpeeseen	Toimitetaan vain ja ainoastaan tarvittavia tuotteita tai tietoa asiakkaalle, vasta silloin kun niitä tarvitaan, ja vain sen verran kuin niitä tarvitaan.
Kaizen	Tarkoittaa parempaan muutokseen pyrkimistä. Käytetään jatkuvan kehittämisen apuna. Tämä prosessi auttaa löytämään hukkatointoja ja poistamaan niitä.
Heijunka	Kysynnän ja tuotantonopeuden tasoittaminen stabiiliksi edellyttää nopeita asetuksia tuotetyypin vaihtuessa.
Visualisointi	Visuaalisen ohjausjärjestelmän kehittäminen. Selkeät signaalit ja laput, jotka sovitun mukaisesti ohjaavat materiaalien liikettä ja täydentämistä tuotantolinjassa
Arvovirtakartat	Arvovirtojen kuvaaminen, miten paljon aikaa menee mihinkin toimintaan prosessin vaiheissa, millaisia puskureita esiintyy prosessin eri vaiheissa.

### 2.1.5 Kanban-kehitysmalli

Kanban ja Lean-ajattelu ovat molemmat lähtöisin Japanista 1950-luvulta. Kanban tarkoittaa japaniksi mainoskylttiä tai taulua. Kanban-menetelmän tarkoitus on aikatauluttaa toimintoja ja pitää työntekeo jatkuvana virtana. Se suunniteltiin Toyotalla toteuttamaan käytännön tarpeita Lean-ajattelun tueksi. Kanban otettiin käyttöön ohjelmistokehitykseen vasta 2000-luvulla Microsoftilla[12]. Ohjelmistokehityksessä Kanban-taulu auttaa visualisoimaan työtehtäviä ja helpottaa töiden allokointia. Kuten kuvasta 5 näkee, kanban-taulusta huomaa yhdellä vilkaisulla työn tilanteen.[13][14]

Kanban ja Lean-ajattelu pyrkivät samoihin lopputuloksiin, joten niistä löytyy paljon yhtäläisyyksiä. Kuten taulukosta 2 nähdään, molemmat pyrkivät jatkuvaan parantamiseen ja hukkatuimintojen vähentämiseen. Kanban auttaa myös rajoittamaan käynnissä olevien työtehtävien määrää projektin jokaisessa vaiheessa. Tällöin työ ei kasaannu ja on paljon helpompi seurata, miten työ etenee. Tämä helpottaa työn seuraamista ja läpimenoaikojen vertailua ja mittaamista. On todettu että työntekijät voivat keskittyä paremmin työhönsä kun heillä on vain muutama tehtävä kerrallaan työpöydällä.[12][13]



Kuva 5: Esimerkki kanban-taulusta

### 2.1.6 Lean startup-malli

Lean Startup-malli esiteltiin 2008 ja sen kehitti Eric Ries. Ries oli toiminut useissa startup-yrityksissä, jossa tavoitteena on saada mahdollisimman nopeasti tuotettua valmiita tuotteita ja kerättyä rahaa sijoittajilta. Lean startup-mallissa käytetään aika alussa palvelun iteratiiviseen kehittämiseen, jossa asiakastarve on mukana heti alusta alken. Tällä tavoin saadaan vähennettyä riskejä tuotteen kehittämisessä ja siinä, että tuotteelle ei löydy kysyntää. Markkinoinnin ja myynnin aloittaminen on helpompaa,

Taulukko 2: Lean- ja kanban-periaatteet.

Lean-periaatteet	Kanban-periaatteet
Vähennä hukkatointintoja	Visualisoi työtehtävät
Panosta laatuun	Rajoita käynnissä olevaa työn määrää
Luo tietoa	Mittaa ja hallitse työn jatkuvuutta
Vältä lukittautumista	Tee prosessikäytännöistä yksiselitteisiä
Toimita nopeasti	Paranna toimintaa yhdessä
Kunnioita ihmisiä	hyödyntämällä hyviä malleja ja metodeita
Optimoi kokonaisuus	

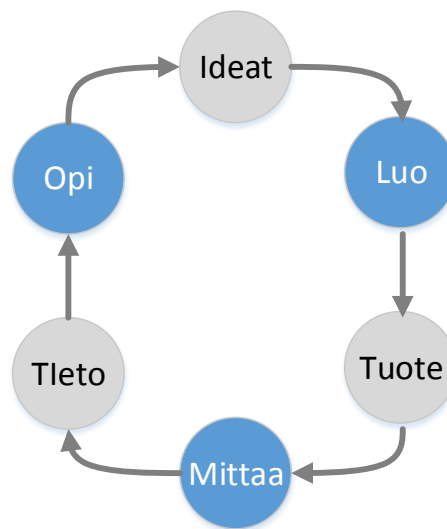
kun on olemassa jo asiakkaita ja kiinnostuneita käyttäjiä. Lean startup-mallia ei ole suunniteltu pelkästään startup-yrityksille vaan Riesin kehitysmalli oli myös valmis käytettäväksi isoissa teknologiayrityksissä. Kehitysmallia voidaan hyödyntää missä tahansa tiimissä oli se sitten startup-yrityksen tai suuren kansainvälisen yrityksen sisällä oleva ryhmä ihmisiä.[15][16]

Lean Startup-menetelmä perustuu tekemiseen, mittaamiseen ja oppimiseen. Kuvassa 6 on esitetty menetelmä yksinkertaisuudessaan. Toiminnassa on tarkoitus mitata jatkuvasti tekemistä, jotta tiedetään miten tulevaisuudessa pitäisi tehdä. Mittaaminen on vain työkalu siihen, että tekijät oppivat jatkuvasti lisää. Oppimista halutaan syntyvän asiakkaista, tuotteesta, tuotealueesta, prosesseista, menetelmistä ja työkaluista. Mittaustulosten ja oppimisen perusteella on helpompi rakentaa seuraavaa tuotetta tai osatuotetta.[15]

Lean startup-filosofia nojautuu perinteisiin Lean-periaatteisiin, joiden tarkoitus on poistaa turhia toimintoja ja lisätä lisäarvoa tuottavia toimintoja tuotekehitysvaiheessa. Hukan poistaminen johtaa siihen, että lean startup-mallia voi toteuttaa hyvin pienellä pääomalla eikä suuria alkusijoituksia tai pitkiä kehitysjaksoja tarvita tuotavien palveluiden ja ideoiden huomiseen. Riesin mallissa hyödynnetään jatkuvasti asiakaspalautetta ja sen tarkoitus on varmistaa, ettei missään vaiheessa käytetä aikaa tuotteen suunnitteluun, joka ei vastaa asiakastarpeeseen. Tuotekehityksessä hyödynnetään liiketoimintamallitauluja (engl. Business Model Canvas) ja useita muita ketterän kehityksen työkaluja. Asiakkaalta kerättävä palaute auttaa myös tuotteen ja liiketoimintamallin määrittelyä. Potentiaalisille kuluttajille ja käyttäjille ei ole tarkoitus mennä vasta valmiin tuotteen kanssa, vaan palautetta kerätään jokaisesta tuotteen osa-alueesta. Lopulta asiakkaalle on tarkoitus kehittää pienin kannattava tuote (engl. Minimum Viable Product), joka tarkoittaa vähimmällä työllä tehtävää tuotetta, joka tyydyttää asiakkaan tarpeet.[15]

Pienimmän kannattavan tuotteen on tarkoitus testata niitä olettamuksia, joita startup-tiimi on kehittänyt ja asiakkaiden palautteesta kerännyt. Pienimmän kannattavan tuotteen kehittämisessä on vielä tärkeää yrittää olla sitomatta kuluja, jotta muutoksia tuotteeseen voidaan tehdä ilman että rahaa menetetään. Tämä menetelmä sopii hyvin vielä hämärän peitossa oleville ja nopeasti muuttuville markkinoille. Mallin sisäänrakennettuun filosofiaan kuuluu, että asiakaspalautteen ollessa ristiriidassa

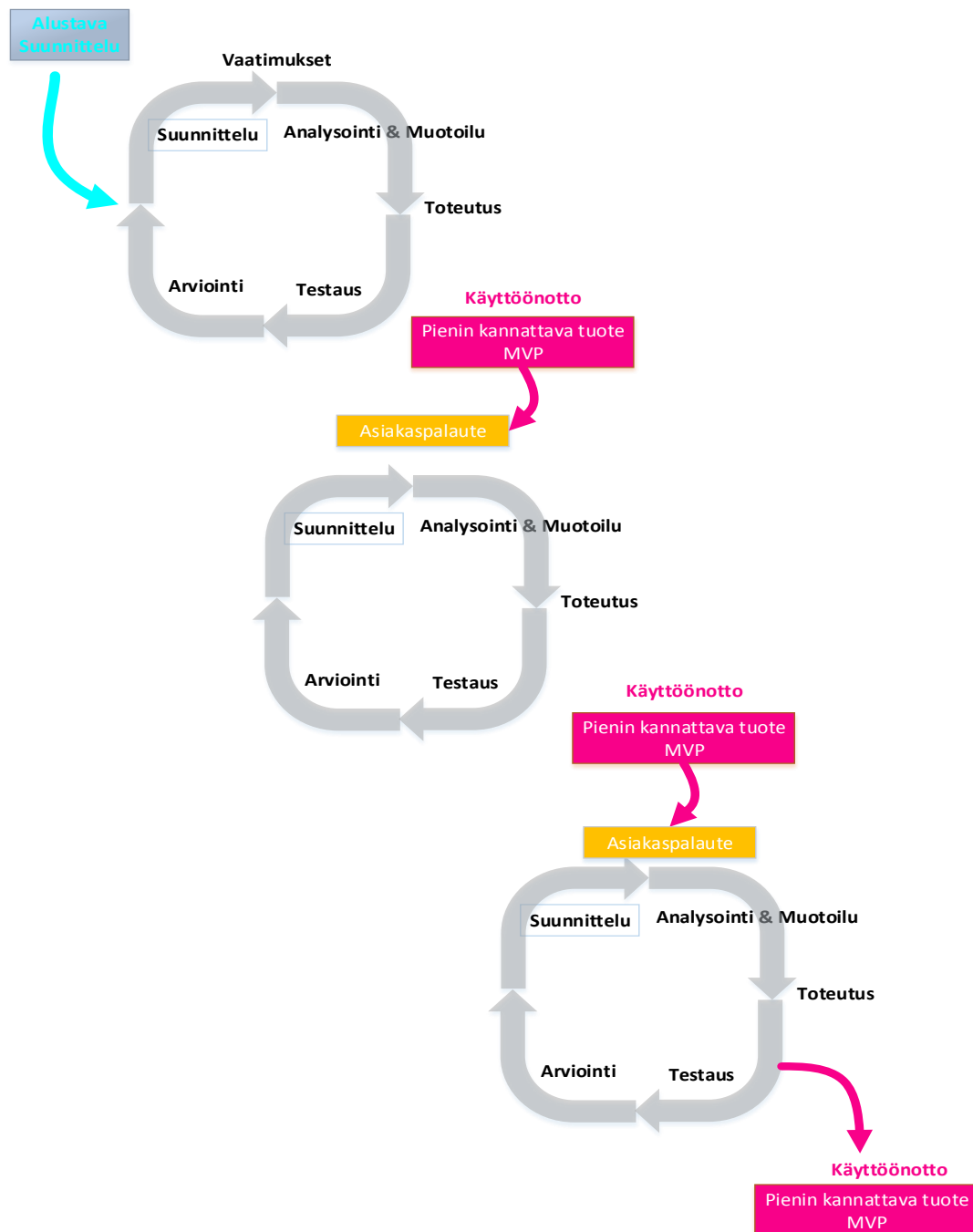
tuotteen kanssa voidaan tehdä käännös (engl. Pivot). Käännöksen tarkoitus on ohjata tuotekehitys täysin eri suuntaan, kun mihin aiemmin on kehitystä suunnattu. Teoriaa tukee useat onnistuneet käännösliekket yrityselämässä, jossa jo hiipumaisillaan oleva yritys on lähtenyt tuottamaan onnistuneesti eri tuotetta. Kehotus muuttaa jo mietittyä mallia antaa vapauksia kokeiluun ja epäonnistumiseen, jotka ovat ehdottomia nopeasti muuttuvilla markkinoilla. Kun tähän yhdistetään lean startup-filosofia, jossa sidottuja kuluja on mahdollisimman vähän, ollaan lähempänä onnistunutta innovaatiota. Usein startup-yritykset epäonnistuvat juuri siksi, että kerättyä rahaa ei ole paljon ja sekin on sidottu ensimmäiseen ideaan ja huomataan, että ratkaisu ei olekaan riittävän hyvä.[15][16]



Kuva 6: Lean Startup-menetelmän ajatus yksinkertaistettuna

Lean startup-malli hyödyntää Lean-periaatteita, ketterää kehitystä, aktiivisesti asiakspalautetta, Startup-yritysten menetelmiä. Samalla malli ei oleta, että tiimi loisi ensimmäisellä kerralla täydellisen liiketoimintamallin, eikä se oleta, että alkuperäinen idea, minkä pohjalta on toimittu, olisi hyvä. Lean startup-malli antaa paljon anteeksi tiimin jäsenille, sillä usein tuotetta ja palvelua lähdetään tuottamaan täysin puhtaalta pöydältä. Ensimmäinen sykli on kuitenkin usein hankalin ja tässä kohtaa asiakspalautteen hyödyntäminen on usein hankalaa, sillä pienin kannattava tuote saattaa olla vielä hyvin keskeneräinen. Ennen pienimmän kannattavan tuotteen valmistamista on tiimin täytynyt käydä tietyt vaiheet läpi, joita toteutetaan jatkuvasti ennen ja jälkeen jokaista tuoteversiota.[15]

Lean startup-mallin kehitysvaiheet ovat: Arviointi, suunnittelu, analysointi ja muotoilu, implementointi ja testaus. Aina vaiheet eivät mene samassa järjestyksessä ja osa vaiheista voidaan toteuttaa samanaikaisesti. On kuitenkin tärkeä huomata, että ensimmäisessä syklissä aloitetaan suunnittelu sen pohjalta mitä aiheesta tiedetään. Tässä on tärkeää hyödyntää viimeisin mahdollinen tieto mitä maailmalla on ja mitä tiimin sisällä asiasta tiedetään. Myöhemmissä sykleissä keskitytään arvioimaan pienimmän kannattavan tuotteen perusteella saatua asiakspalautetta.



Kuva 7: Lean startup-mallin syklisyys

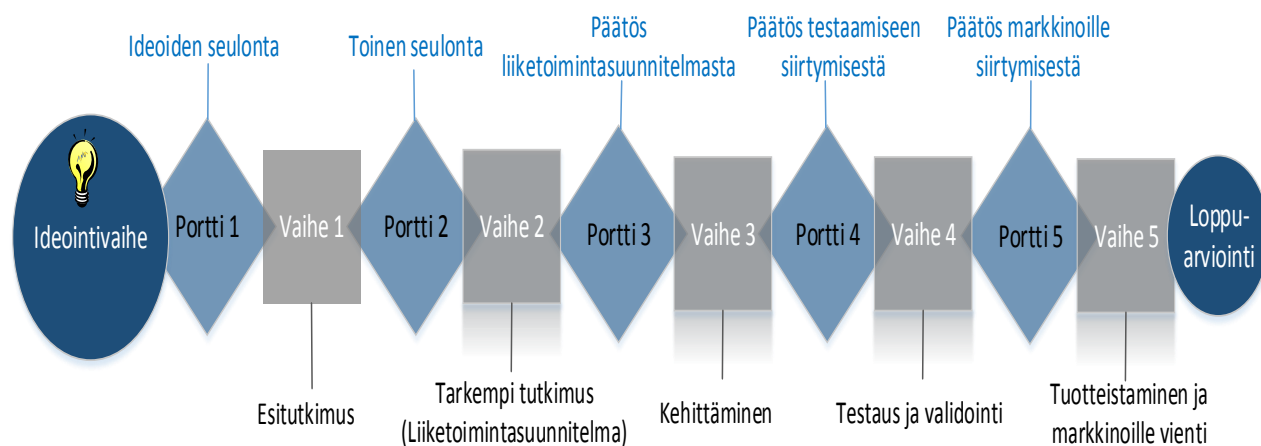
Lean startup-mallilla toimittaessa voidaan lähteä liikkeelle ajatuksesta, että tuote ei ole koskaan valmis. Kuvassa 7 näkyy hyvin Lean Startup-mallin syklinen eteneminen. Jos yrityksellä on varaa kehittää tuotettaan näin perusteellisesti ja tiimi koostuu asiantuntevista yksilöistä, niin tuotteen pitäisi vastata aina asiakkaiden tarpeita.[15][16]

## 2.2 Perinteiset tuotekehitysmallit

Perinteisinä tuotekehitysmalleina tämän tutkimuksen yhteydessä pidetään 1970- ja 80-luvulla esitelty vaihe-portti-malli (engl. Stage-gate) ja vesiputousmalli, jotka ovat vielä monien yritysten tuotekehitysprosessien taustalla.[17]

### 2.2.1 Vaihe-portti-malli

Vaihe-portti-kehitysmallin luoja pidetään Cooperia, joka on tehnyt useita julkaisuja aiheesta 1980-luvulta asti[17]. Tarkistuspisteiden tai porttien käyttäminen ei ollut asiana uusi, mutta 1980-luvulla tämän prosessin luominen uudelleen sai aikaan tehokkaamman mallin. Tämän mallin oli tarkoitus helpottaa innovaatioiden johtamista ja tuotekehitysprosessin hallitsemista.[17]



Kuva 8: Vaihe-portti-kehitysmalli

Vastaavan tyyppisiä lineaarisia kehitysmalleja on useita ja nimiäkin kehitysmalleille löytyy useita. Vaihe-portti-konsepti rakentuu innovaatioprosessin johtamiseen. Prosessi on jaettu useaan eri vaiheeseen tai osa-alueeseen. Jokaisen vaiheen välissä on portti tai tarkistuspiste, jossa on määritellyt kriteerit, jotka tuotteen tai palvelun täytyy täyttää. Itse kehitystyö tehdään vaiheissa ja portit varmistavat, että tuote on tarpeeksi laadukas, jotta seuraavaan työvaiheeseen voidaan siirtyä. Innovaatioprosesseissa tuote tai palvelu on jaettu usein ennalta määrättyihin vaiheisiin, joita on tyypillisesti neljästä seitsemään kappaletta. Kuvassa 8 nähdään tyypillinen vaihe-portti-mallin kehitysprosessi portteineen ja vaiheineen.[17]

Ennen siirtymistä uuteen vaiheeseen on täytettävä porttiin kirjatut vaatimukset. Vaatimuksiksi voidaan asettaa tietoja, toiminnallisuuksia tai tuotoksia, joita projektin tulee sisältää, jotta seuraavaan vaiheeseen voidaan jatkaa. Kun palvelun tai projektin viimeisen vaiheen tuotoksia tutkitaan portilla, sille voidaan joko antaa kyllä-, ei- tai odota-päätös. Yleisesti ottaen projekteissa projektipäällikkö vastaa siitä, että portille tultaessa projektille asetetut vaatimukset täyttyvät ja voidaan siirtyä eteenpäin. Portti voi vielä projektin alkutaipaleella olla automaattinen tarkistuslista, mutta tarvittavien resurssien kasvaessa myöhemmässä vaiheessa portilla toimii usein ryhmä ylemmän tason johtajia portinvartijoina. Portilla katsotaan jo suoritettun vaiheen laatua, mutta tarkastellaan myös seuraavan vaiheen suunnitelmaa, jotta resursseja projektille voidaan myöntää. Vaihe-portti-malli vaatii organisaatiolta järjestystä, jossa projektia vetää projektijohtaja alusta loppuun, eikä itse projektia siirretä osastolta toiselle.[17]

Vaihe-portti-mallin on todettu parantavan tuotekehitysprosessia 1980-luvulla verrattuna epämuodollisempaan ja heikommin johdettuun malliin[18]. Mallin hyvyys perustuu kuitenkin paljon porttien vaatimuusmäärittelyyn, sekä joissakin portinvartijoina toimivien ylempien johtohenkilöiden ammattitaitoon ja sitoutumiseen. Suurin osa uuden tuotteen kehitysprojekteista ei pääse ikinä siihen vaiheeseen, että tuote kaupallistettaisiin. Eräässäkkin raportissa on todettu, että ideointivaiheessa yksi seitsemästä projektista tulee menestymään kaupallisesti[18]. Tämän vuoksi on tärkeää, että portteihin on sisäänrakennettu riittävän hyvät arviointimekanismit ja niitä noudatetaan. Kun tuotteen tai palvelun etenemistä seurataan porteilla, on tärkeää arvioida milloin projekti näyttää siltä, että on parempi lopettaa se kuin antaa myöntävä päätös.[17]

Kuvan 8 esimerkkimallin mukainen palvelukehitysprosessi on hyvin tyypillinen ja käytetty malli useissa organisaatioissa. Mallin hyviin puoliin kuuluu ennustettavuus ja läpinäkyvyys. Projektipäällikkö tietää ideointivaiheen jälkeen ensimmäisen vaiheen suoritettuaan minkälainen projektista tulee ja miten paljon resursseja sitä varten tulee varata. Tämä tietenkin tarkentuu tarkemmassa liiketoimintamallin kuvauksessa seuraavassa vaiheessa, mutta suuntaa antavan määrittelyn jälkeen tiedetään jo mitä aiotaan tehdä ja milloin. Projektiportfolion hallinta on myös helpompaa ja yksinkertaisempaa, kun projekteja voidaan vertailla helposti keskenään, ja nähdään missä vaiheessa ne etenevät. Myös jokaisen portin sisältämät vaatimukset ja vaiheiden tavoitteet ovat kaikille selvät, joten tiedetään alusta alkaen mitä pitää tehdä. Kehitysmallin etuna ovatkin parempi päätöksenteko, syvempi keskittyminen tiettyyn vaiheeseen ja vähemmän laatuvirheitä.[17]

### 2.2.2 Vesiputousmalli

Winston Royce esitti Vesiputous-mallin ensimmäisen kerran 1970-luvulla[19]. Vesiputousmalli luotiin alun perin ohjelmistokehitystä varten. Siitä huolimatta se on otettu käyttöön useiden yritysten tuotekehityksessä. Kehitysmalli on hyvin samankaltainen prosessi kuin portti-vaihe-malli sen lineaarisuuden vuoksi. Perinteisessä

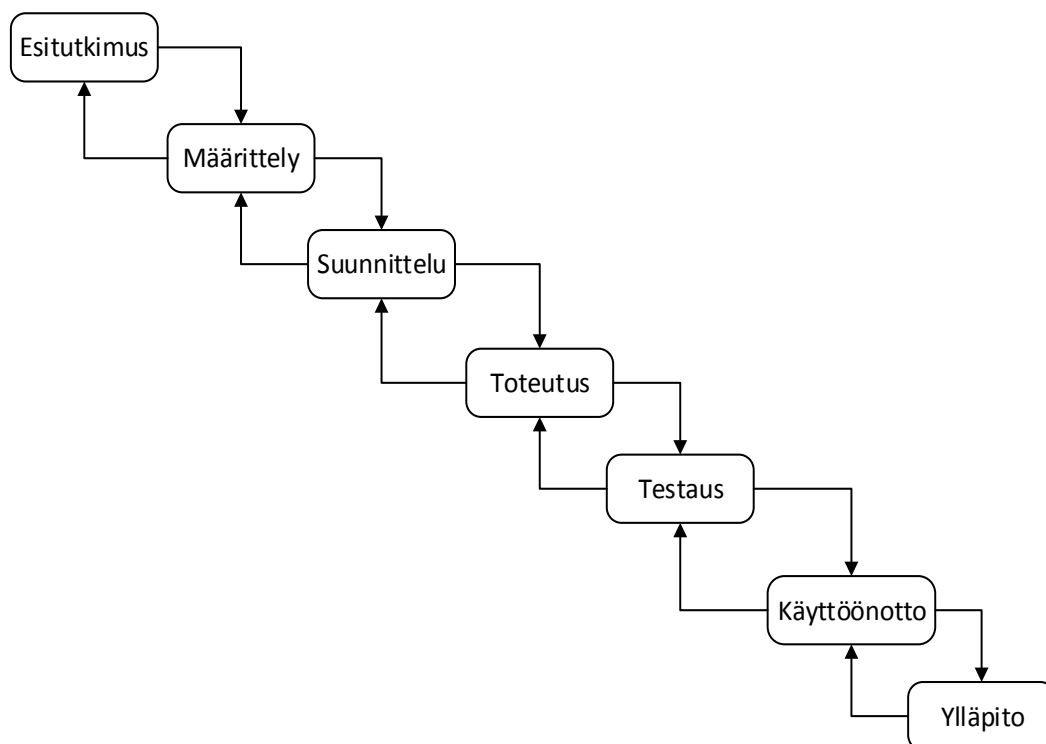


vesiputousmallissa edellinen vaihe suoritetaan loppuun ennen kuin seuraava voidaan aloittaa.

Vesiputousmallissa projekti aloitetaan tutkimalla aihetta ja edetään kuvan 9 mukaan vaiheittain suunnittelusta testaukseen. Ensin suunnitellaan kokonaisuutta ja tarvittavia järjestelmiä, joita kehitettävä tuote tai palvelu tarvitsee. Usein suurissa projekteissa tämä vaihe on tärkein, sillä se sitoo kuluja ja muutokset tämän pisteen jälkeen ovat kalliita ja hankalia toteuttaa. Järjestelmäsuunnittelun jälkeen vuorossa ovat ohjelmistot ja niiden vaatimusten määrittely. Tässä kohtaa mietitään myös, miten eri ohjelmistot keskustelevat keskenään, jotta hyöty välittyy loppukäyttäjälle. Ohjelmistojen jälkeen vuorossa on arkkitehtuurinen suunnittelu, joka yhdistää edelliset kaksi vaihetta ja miettii miten eri järjestelmät ja ohjelmistot keskustelevat keskenään ja mitä rajapintoja pitää rakentaa. Tämä on ensimmäinen kohta jossa määritellään tarkemmin miten eri järjestelmät ja ohjelmistot ovat keskenään yhteydessä. Viimeisenä suunnittelun vaiheena on yksityiskohtainen suunnittelu, jossa jokainen yksittäinen tekijä suunnitellaan ja määritellään. Tämän jälkeen ohjelmistoprojekteissa on vuorossa koodaaminen ja palvelukehityksessä sitä voidaan kutsua toteutusvaiheeksi. Kun suunnitelman mukainen malli on valmis, voidaan siirtyä testaukseen, jonka tarkoitus on kokeilla, vastaako tuote tai palvelu sen määrittelyä ja löytyykö siitä virheitä. Testauksen jälkeen tuotos voidaan siirtää ylläpidon hoidettavaksi, kun mahdolliset löytyneet virheet on korjattu ja palvelu toimii. [20]

Vesiputousmalli on hyvä palvelukehityksessä silloin kun tunnetaan ongelmakenttä ja ratkaisu voidaan määrittellä yksiselitteisesti projektin alkuvaiheessa. Laadunhallinta on helpompaa, kun palaset voidaan rakentaa yksi kerrallaan luotujen suunnitelmien perusteella. Vesiputousmallissa tehtävät muutokset projektin loppupuolella ovat usein kalliita ja aikaa vieviä, joka korostaa alussa tehtävän määrittelyn tärkeyttä. Tätä mallia käyttäessä vältetään turhalta tekemiseltä ja riskeiltä, sillä suunnittelun aikana tehtävät muutokset eivät vielä maksa mitään eikä aikaa mene hukkaan asiakaspilotoinneissa keskeneräisin tuottein. Tuotekehitysprojektissa ei myöskään tarvitse olla niin tiivistä kanssakäymistä ryhmän jäsenten kesken, vaan projektipäällikkö voi hoitaa tuotekehityksen suhteet sidosryhmien kanssa. Vesiputousmallin käyttö kuitenkin vaatii asioiden valmista ymmärrystä, sekä kykyä hahmottaa järjestelmään kuuluvia ohjelmistoja ja rajapintoja. [20]

Vesiputousmalli on ollut jo niin pitkään käytössä eri yrityksissä, että siitä on muokattu useita eri johdannaisia. Usein tuotekehitysprojekteissa pyritään läpimenoajan nopeuttamiseen, jolloin lineaarinen malli ei ole optimaalisin menetelmä tuotekehityksessä. Tämän vuoksi vesiputousmallia on muokattu niin, että alkupään vaatimusten määrittelyä ja järjestelmäsuunnittelua voidaan tehdä päällekkäin. Isoissa yrityksissä on usein erilliset IT-arkkitehdit, jotka vastaavat järjestelmistä ja ohjelmistoista, joten eri määrittelyvaiheita voidaan tehdä samanaikaisesti. Kokonaisprojekti voidaan myös jakaa pienemmiksi osaprojekteiksi, jolloin läpimenoajat lyhenevät ja kokonaisuuden hahmottaminen onnistuu vielä muutamalta ihmiseltä. Muokatut vesiputousmallit ovatkin joustavampia eivätkä sido samoja ihmisiä todella pitkäaikaisiin projekteihin. Tällöin myös pienempien osa-alueiden toteutus nopeutuu ja koko projektin



Kuva 9: Vesiputous-kehitysmalli

valmiusprosentti kasvaa nopeammin.[20]

## 2.3 Perinteisten tuotekehitysmallien ero ketteriin

Aikaisemmissa luvuissa on esitelty useita eri ketteriä palvelukehitysmalleja sekä tuotu myös perinteisen mallien käytäntöjä. Tämän luvun tarkoitus on tiivistää erot konkreettisella tasolla, sekä esittää kvalitatiivisia poikkeamia näiden menetelmien välillä.

### 2.3.1 Kehitysmallien erot teoreettisella tasolla

Teknologioiden kehittyessä ja kilpailun kasvaessa yritykset ovat törmänneet epävarmuuden kasvuun. Kyky reagoida jatkuvasti kiihtyvään muutokseen on vaatinut yrityksiltä uudenlaisia toimenpiteitä. Näitä toimenpiteitä on kutsuttu ohjelmistokehityksessä ketteräksi kehittämiseksi. Perinteisten mallien raskaat suunnitteluvaiheet ja muutosten kustannukset ovat antaneet aihetta kehittää uusia lähestymistapoja tuotekehitykseen. Alla olevissa taulukoissa 3 ja 4 on esitelty pääpiirteittäin ketterien ja lean-kehitysmallien erot perinteisiin. Molemmissa suuntauksissa suurimmat erot ovat asiakkaan roolin tärkeys sekä muutosten ja epäonnistumisten ennakoiminen. Uudet mallit ovat sekä epäformaalimpia että autonomisempia. Uusissa malleissa

valta siirtyy prosesseilta ihmisille ja rationalististen arvojen sijaan empiristiset arvot nousevat tärkeämmiksi.[21]

Taulukko 3: Ketterän ja perinteisen kehitysmallien erot

<b>Ketterä</b>	<b>Perinteinen (vesiputous)</b>
<b>Johtaminen ja organisointi</b>	
Ihmisojohtaminen	Kontrolloitu johtaminen
Joustava	Kurinalainen / ennalta määrätty
Johtaja toimii fasilitaattorina	Johtaja toimii suunnittelijana
Hyödynnetään hiljaista tietoa	Nojaututaan eksplisiittiseen tietoon
Palkitseminen tiimitasolla	Palkitseminen yksilötasolla
<b>Ihmiset</b>	
Tehdään yhdessä tiimin kanssa	Itsenäinen tekeminen
Monialainen osaaminen	Erikoistunut osaaminen
Osallistava ja yhteinen päätöksenteko	Johto tekee päätökset
Asiakkaan vahva osallistaminen	Asiakkaan vähäinen osallistaminen
Pienet ydintiimit	Isot tiimit
<b>Prosessit</b>	
Ihmiskeskeiset toimintatavat	Prosessikeskeiset toimintatavat
Kokeilevat toimintatavat	Standardoidut toimintatavat
Testataan ennen kuin tehdään	Tehdään ennen kuin testataan
Joustava tekeminen	Ennalta suunniteltu tekeminen
Iteratiivinen	Lineaarinen
Lyhyet toimintajaksot	Pitkät toimintajaksot

[22]

### 2.3.2 Kehitysmallien suorituserot projekteissa

Oikein implementoituna ketterät kehitysmenetelmät parantavat laatua ja kasvattavat tuotteen tai palvelun arvoa enemmän kuin perinteiset menetelmät[23]. Standish Group on julkaissut yli 20 vuotta projektien onnistuvuutta ja tällä vuosikymmenellä useissa julkaisuissa on pohdittu ketterien ja vesiputousmallien onnistuvuuksia. 2015 raportissa on tutkittu 50000 projektien onnistuvuusprosentteja projektin koon ja kehitysmallin mukaan[24]. Tämän tutkimuksen tulokset ovat julkaistu taulukossa 5. Tutkimuksen tuloksena nähdään, että ketteryys kasvattaa onnistuvuutta huomattavasti sitä enemmän, mitä suuremmasta projektista on kyse.

Taulukko 4: Lean-startup-mallin ja perinteisen kehitysmallien erot

Lean-startup	Perinteinen (vesiputous)
<b>Strategia</b>	
Liiketoimintamalli Hypoteesiin perustuva	Liiketoimintasuunnitelma Toteutukseen perustuva
<b>Kehittäminen</b>	
Ketterä kehitys Rakenna tuote iteratiivisesti ja vähitellen	Ketterä tai perinteinen kehitys Rakenna tuote iteratiivisesti tai määrittele etukäteen ennen rakentamista
<b>Uuden tuotteen luomisprosessi</b>	
Asiakaskehitys Lähde ulos toimistolta ja testaa hypoteeseja	Tuotekehitys Tuotteista markkinointia varten lineaarisesti asia kerrallaan suunnitelmien mukaan
<b>Nopeus</b>	
Todella nopea Toimitaan riittävän tiedon pohjalta	Tasainen nopeus Toimitaan täyden tiedon pohjalta
<b>Organisaatio</b>	
Asiakas ja ketterät kehitystiimit Palkataan oppimista, ketteryyttä ja nopeutta varten	Funktionaaliset osastot Palkataan kokemuksen ja toteutuskyvyn perusteella
<b>Epäonnistuminen</b>	
Odotettu Korjataan iteroimalla ideoita ja tehdään käännös (engl. Pivot) toimimattomista konsepteista	Poikkeuksellinen Korjataan antamalla potkuja tai vaihtamalla henkilöitä
<b>Talousraportointi</b>	
Käytetään toimivia mittareita Asiakkaan hankintakulu, asiakkaan elinkaariarvo, vaihtuvuus (engl. churn), elinvoimaisuus	Taloudellinen kirjanpito Tuloslaskelma, tase, rahavirtalaskelma

Taulukko 5: Ketterien ja perinteisten projektien onnistuvuus

Koko	Metodi	Onnistunut	Kyseenalainen	Epäonnistunut
Kaikenkokoiset projektit	Agile	39 %	52 %	9 %
	Waterfall	11 %	60 %	29 %
Suuret projektit	Agile	18 %	59 %	23 %
	Waterfall	3 %	55 %	42 %
Keskikokoiset projektit	Agile	27 %	62 %	11 %
	Waterfall	7 %	68 %	25 %
Pienet projektit	Agile	58 %	38 %	4 %
	Waterfall	44 %	45 %	11 %

## 2.4 Ketterän palvelukehitysmallin skaalaaminen isossa yrityksessä

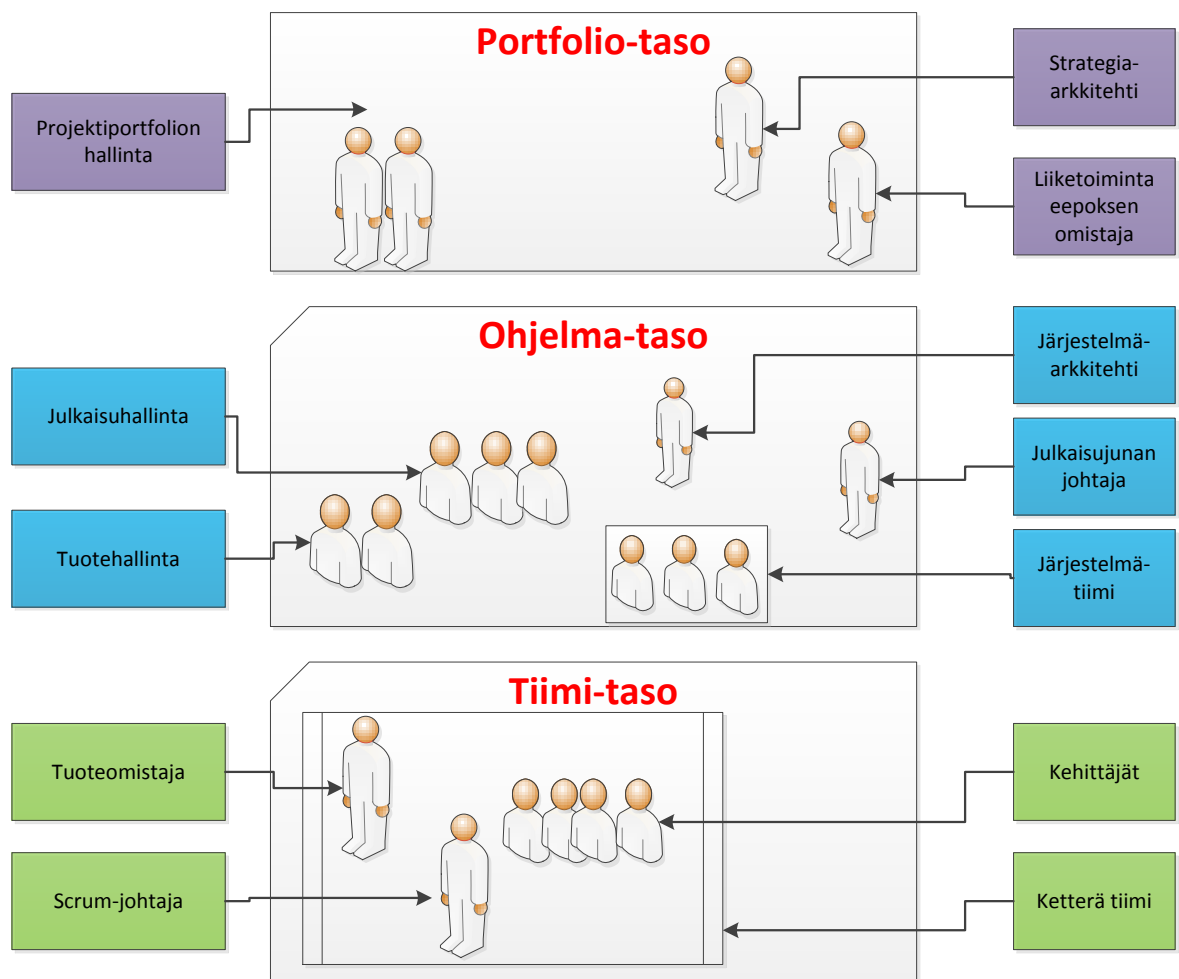
Scrum-kehitysmalli keskittyy optimoimaan yhden tiimin tekemistä yhdessä tuotteessa tai palvelussa. Isoissa organisaatioissa Scrum-tiimejä on kuitenkin niin monia, että niitä varten on oltava hallittavissa oleva johtamismenetelmä. Kun saman tuotteen tai palvelun parissa on useampi Scrum-tiimi, se vaatii hyvää synkronointia kaikkien näiden sidosryhmien välillä. Tilanne mutkistuu entisestään, kun samalla tuotealueella toimii useampi palvelukehitysprojekti, joilla on päällekkäisiä tavoitteita. Ketterän kehittämisen skaalaamista koko organisaation laajuista tekemistä varten on kehitetty useita toimintamalleja, joista tässä luvussa tullaan esittelemään viimeisimmät ja edelleen aktiivisessa kehityksessä olevat mallit.[25]

### 2.4.1 SAFe-kehitysmalli

SAFe (engl. Scaled Agile Framework) on Dean Leffingwellin luoma malli ketterän kehityksen skaalaamiseen organisaatioissa. Malli on jaettu kolmeen kerrokseen, jossa jokaista kerrosta johdetaan tietyn periaattein. Malli on hyvin yleispätevä ja sitä voidaan soveltaa suuriin ja pieniin organisaatioihin. Kuvassa 10 näkyy hyvin mallin hierarkisuus. SAFe pyrkii hyödyntämään Lean-menetelmiä ja ketteriä kehitysmenetelmiä sisällyttäen niiden ajatusmallit, periaatteet ja arvot.[26][27]

Alimmalla tasolla SAFe pyrkii toteuttamaan ketterää kehitystä itseohjautuvien ja poikkifunktionaalisten tiimien avulla. SAFe 4.0-mallissa ehdotetaan tiimin käyttävän Scrum-, XP-, Kanban-mallia tai jotain näiden yhdistelmiä. Käytännössä mallissa ehdotetaan Scrum-mallin tyyppistä toteutusta, jossa tiimit koostuvat 5-10 henkilöstä. Käytettävä malli kuitenkin on rakennettu Agile Manifesto-julkaisun pohjautuviin päätelmiin, jolloin tiimin tulisi toimittaa kahden viikon välein toimivaa ohjelmistoa. Scrum-tiimillä on aina johtaja, ja SAFe suosittelee yhden Scrum-johtajan toimivan muutamassa tiimissä, jotta henkilöresurssit riittävät. Tiimit ovat itseohjautuvia ja valitsevat lopullisen toteutustavan, mutta ryhmällä on kuitenkin oltava omistaja ja johtaja, jotka kommunikoivat ylemmille tasoille. Tiimeille tulee suunniteltuna tehtävälistä ohjelma-tasolta ja tiimille jääkin tehtävien jaksottaminen ja toteuttaminen kahden viikon sykleihin.[28][15][27]

Keskimmäinen taso SAFe-mallissa on ohjelmataso, jossa tiimit ja muut resurssit kootaan yhteen. Yhteistä tavoitetta ajavaa kokonaisuutta kutsutaan ketteräksi julkaisujunaksi (engl. Agile Release Train) ja sen idea on osoittaa sekä tiimeille että ylemmälle johdolle nykytilanne ja tulevaisuuden suunta. Ketterän kehitysjunan tehtävä on myös tuottaa arvoa ja valmiita julkaisuja tuotteisiin tai palveluihin. Jokaisessa junassa on 5-12 itsenäistä ketterää kehitystiimiä, joista jokainen kykenee julkaisemaan testattua ja toimivaa ohjelmistoa. Ketterää kehitysjunaa hallitaan useista näkökulmista, joten junan hallintaa varten on olemassa johtajan, teknologia-arkkitehdin ja tuote- tai ohjelmapäällikön roolit. Näiden tiimien tai henkilöiden avulla pystytään



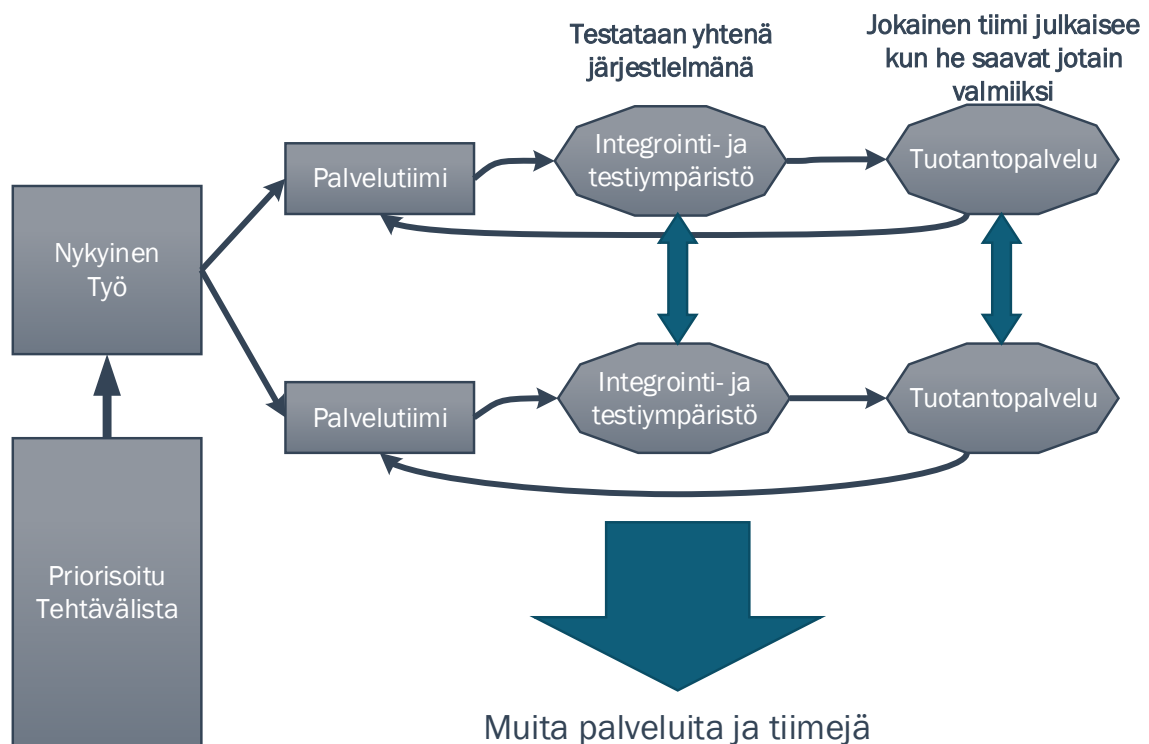
Kuva 10: SAFe-skaalausmalli ketterään kehittämiseen

johtamaan koko kehitysjunan suuntaa ja tarpeita, joita yksittäisistä tiimeistä nousee esille.[28]

Ylimmällä tasolla SAFe toteuttaa projektiportfoliohallinnan näkökulmasta ketterän organisaation johtamista. Ylin johto ja projektiportfoliohallinta vastaa strategisista valinnoista ja linjauksista liiketoiminnalle, sekä se vastaa ohjelmille valituista budjeteista ja tuotelistauksista. Kokonaisbudjetti jaetaan arvovirroille, joita pyritään ohjaamaan liiketoimintaepoksilla (engl. Business Epic). Nämä tarinat tai eepokset ovat ison kuvan strategisista linjauksista, joihin myöhemmin tiimit kirjoittavat käyttäjäkertomuksia. Liiketoimintakertomukset ovat siis liiketoimintamahdollisuuksia, joissa uutta liiketoimintaa voidaan harjoittaa. Näitä liiketoimintamahdollisuuksia varten vaaditaan vielä tekniset valmiudet, joita SAFe mallissa on liiketoimintaeposten välillä. Työn virtauksen säilyttämiseksi SAFe hyödyntää portfoliohallinnassa Kanban-metodia, jossa tuodaan näkyväksi, mitä asioita ja missä järjestyksessä organisaation on tarkoitus toteuttaa.[28][27]

## 2.4.2 MAXOS-kehitysmalli

MAXOS-kehitysmalli (engl. Matrix of Services) on palvelukehityksen viimeisin julkisuuteen tuotu ketterän kehittämisen skaalausmalli. Sen tarkoitus on olla ketterämpi ja nopeampi kuin SAFE-malli ja Scrum-kehitysmalli. Mallin alkuperäinen tarkoitus on ollut julkaista nopeammin palveluita ja parannuksia kuin kilpailijat, jotka toteuttavat aikaikkunaa hyödyntäviä kehitysmalleja kuten Scrum. MAXOS pohjautuu jatkuvaan ketteryyteen ja jatkuvaan kehittämiseen, jonka tarkoitus on jatkuvasti parantaa ja kehittää palveluita ja tuotteita ilman turhia taukoja. Jatkuvan kehityksen idea on, että heti, kun jokin osa-alue on valmis julkaistavaksi, se siirretään tuotantoon. Tämä tarkoittaa, että, julkaisuja voidaan tehdä useita päivässä. Tällöin ei tarvitse odottaa ennalta määrättyä kahden viikon välein tapahtuvaa testausta tai tarkastelua, jossa käsitellään useamman tiimin aikaansaannokset ennen tuotantoon siirtämistä.[29]



Kuva 11: Skaalaaminen MAXOS-kehitysmallin mukaan

Jatkuvaa kehitystä hyödyntäviä yrityksiä on mainittu olevan muun muassa Spotify, Facebook ja Amazon, jotka kehittävät jatkuvasti tuotteitaan ja julkaisevat niistä ohjelmistoparannuksia ilman erillisiä julkaisuaikataulua. Amazonilla on yli 1000 palvelutiimiä, jotka rakentavat ja muokkaavat koodia jatkuvasti ja he muuttavat ohjelmistoaan keskimäärin 12 sekunnin välein. Jokaisella tiimillä on täysi vastuu suunnittelusta, ohjelmoinnista, testauksesta ja julkaisemisesta. Kuvassa 11 nähdään mallin yksinkertaisuus, joka on hyvin samankaltainen kuin Scrum, mutta paljon yksinkertaisempi kuin SAFe. Tämän kehitysmallin käyttäminen vaatii jatkuvaa



monitorointia, sillä jos jokin menee pieleen, niin ilmoituksen täytyy tulla automaattisesti sekunneissa tai minuuteissa. Jatkuvan kehityksen hyödyntäminen vaatii usein kehitys-tuotanto-tiimiä (engl. DevOps), jossa tuotantohenkilöstö pystyy valvomaan ja mahdollistamaan tuotteen tai palvelun toiminnan. Pilvipalvelut mahdollistavat automatisoidut työkalut siihen, että tuotantoon voidaan jatkuvasti integroida uusia toiminnallisuuksia, joita kehitystiimit luovat.[29]

MAXOS on tällä hetkellä vielä niin uusi malli ja käytössä vain niin harvassa yrityksessä. Siitä ei ole olemassa virallista opetusohjelmaa. SAFe myös mahdollistaa hyvin samankaltaisen tekemisen, eikä sulje pois tiimin mahdollisuutta vaikuttaa käytössä oleviin työkaluihin. Tiimi voi käyttää jatkuvaa kehitystä ja kehitys-tuotanto-tiimi voidaan perustaa siitä huolimatta, että taustalla on SAFe-malli. MAXOS-mallin käyttäminen vaatii kuitenkin vahvaa integroitumista pilvipalveluihin, joten niiden yleistyessä myös MAXOS-malli tai jatkuvan ketterän kehittämisen metodologiat ja käytännöt alkavat yleistyä.[29]

Suurin ero SAFe- ja MAXOS-mallin välillä on tiimissä käytettävät metodit ja tiimien muodostaminen. Kuten aiemmin mainittiin, niin SAFe-mallissa käytetään usein Scrum-kehitysmallia, kun MAXOS pyrkii hyödyntämään ketteriä palvelutiimejä, kuten Lean startup-mallissa. Scrum-malli vaatii aina 5-10 henkilön ryhmän, jossa on jokaisen osa-alueen asiantuntijat mukana. Palvelutiimi voi koostua 2-8 henkilöstä ja siihen voidaan rekrytoida tarvittaessa ulkopuolisia, mutta kehittäminen voidaan aloittaa heti. Scrum vaatii johtajan, omistajan, kehittäjät ja asiantuntijat ja heidän pitää sopia useista asioista ennen kuin työtä voidaan aloittaa. Palvelutiimi voi vastaanottaa tehtäviä ja muutoksia milloin vain, mutta Scrum-tiimillä on usein kahden viikon odotusaika ennen kuin mitään pyyntöjä ja muutoksia voidaan toteuttaa. SAFe-malli vaatii usein isojen ryhmien kokoontumisia, kun ylemmän tason johtajat sopivat asioita ja tekevät suunnitelmia. Nämä tapaamiset ovat ennalta sovittuja ja kuormittavat organisaatioita, sillä voi olla, että tarvetta tapaamiselle ei ole juuri silloin ja välillä tarvetta olisi useammin. MAXOS palvelutiimit kehittävät ja julkaisevat tuotantoon jatkuvasti muutoksia, kun Scrum-tapaamiset järjestetään kahden viikon välein, jolloin julkaisu voidaan tehdä.[29]

## 2.5 Ketterän palvelukehityksen johtaminen

Johtamista voidaan tutkia useiden eri osa-alueiden pohjalta ja useissa eri tasoissa. Johtamista voidaan toteuttaa strategian johtamisesta aina yksilön itsensä johtamiseen. Tässä työssä on olennaista keskittyä palvelukehitysprosessin näkökulmasta siihen eniten vaikuttaviin osa-alueisiin. Ihmisjohtaminen on olennainen osa ketterän kehityksen johtamisessa, koska ketteryyden peruseriaatteisiin kuuluu ihmisten asettaminen prosessien edelle. [30]

### 2.5.1 Ketterän johtamisen onnistuvuus

Ketterät menetelmät nostavat ihmiset prosessien edelle, tuotokset dokumentoinnin edelle, asiakasyhteistyön sopimusneuvotteluiden edelle ja muutosten tekemisen suunnitelmassa pysymisen edelle. Nämä ohjenuorat ohjaavat kehittämisen pieniksi paloiksi, jolloin kokonaisuuksia voidaan tehdä ketterästi huomioiden asiakkaan tai markkinoiden muutokset. Nämä kaikki osatekijät vaikuttavat siihen, että kaikkea ei kannata suunnitella etukäteen hyvin tarkalla tasolla eikä yhtä tiettyä toimintamallia kannata toteuttaa projektin alusta loppuun. Nämä vaatimukset koskevat myös projektipäällikköä tai johtajaa, jolloin tiukka ja hallitseva johtaja ei sovellu ketteriin kehitysprojekteihin. Eri johtamistyylien vertailu ei anna riittävästi tietoa siitä, mitä vaaditaan hyvältä ketterältä johtajalta.[31]

Ihmisten johtaminen voidaan jakaa kahteen eri johtamistyyppiin, jotka ovat transaktionaalinen ja transformationaalinen. Transaktionaaliset johtajat käyvät vaihtokauppaa suoritusten ja palkkioiden avulla. Tässä mallissa johtaja yrittää saavuttaa mahdollisimman paljon etuja itselleen tai ryhmälleen eikä päämäärä ole saavuttaa yhteisiä etuja. Transformationaalisessa mallissa johtajan tehtävä on selvittää alustensa tarpeet, motiivit, vaatimukset ja heidän kokonaispersoonallisuutensa. Tässä mallissa korostetaan johtajan ja alaisten välistä vuorovaikutusta, jonka tarkoitus on herättää sisäistä motivaatiota ja innostaa alaisiaan suoriutumaan tehtävistään. Transformationaalisessa johtajuudessa usein sekä alaiset että johtaja kehittyvät säännöllisen kommunikoinnin avulla. Nämä mallit eivät ole kuitenkaan toistensa vastakohtia vaan voivat myös täydentää toisiaan, jos transaktionaalinen johtamismalli nähdään enemmän asioiden johtamisena[32]. Vaikka transaktionaalinen ja transformationaalinen johtamismalli eroaa toisistaan hyvin paljon, niissä hyödynnetään esimiehen valtaa. Tämän takia molemmat mallit ja soveltuvat perinteisiin organisaatioihin, joissa on hierarkiatasoja.[33][34]

Asiakkaiden tulisi olla sitoutuneita projektiin, riittävän asiantuntevia, yhteistyökykyisiä ja heillä pitäisi olla riittävä valtuutus, jotta lopputulos tuottaisi mahdollisimman paljon lisäarvoa. Koska ketterässä kehityksessä dokumentoinnin osuus ei ole tärkeintä, niin on ehdottoman tärkeää saada kaikki asiakkaan hiljainen tieto siirrettyä tuotokseen. Tämä korostaa transformaation johtamisen roolia ja ihmisten välisen kommunikoinnin tärkeyttä.[35]

Johtajien persoonallisuuksia on tutkittu pitkään, mutta ketterän johtajan hyviä ominaisuuksia suhteessa hänen ajattelutapaansa on tutkittu vasta vähän. MBTI (engl. Myers Briggs Type Indicator) on yksi yleinen tapa arvioida henkilön persoonallisuutta. Tätä tapaa on tutkittu ja käytetty hyvin usein ryhmädynamiikan ja tiimien muodostamiseen ja johtamiseen. Bonner on kuitenkin tutkimuksessaan todennut, että johtajan kykyä toimia ketterässä ympäristössä varten on oltava olemassa muita arviointitapoja[31]. Yksilön tapa ajatella antaa tarkemman kuvan siitä, miten hän toimii ja miten hän hyödyntää tietoa. Bonnerin tutkimuksessa on analysoitu Churchmanin viiden tiedonhankintamenetelmän pohjalta, miten johtaja suoriutuisi ketterässä ympäristössä[31]. Churchmanin viisi tiedonhankintamenetelmää kuvaa

yksilön kykyä ajatella asioista ja hän luokittelee johtajat viiteen kategoriaan, jotka ovat eritelty tarkemmin taulukossa 6.[34]

Ihmiset voidaan jakaa näihin kategorioihin suorittamalla InQ-testi (engl. Inquiry Mode Questionnaire) joka kertoo ihmisen tavasta ajatella johtajana. Testi itsessään on suuntaa antava, mutta sen yhdistäminen tosielämän tilanteisiin antaa jo tarkemman kuvan henkilöstä. Taulukosta 7 voidaan nähdä, miten ihmisen ajattelutapa vaikuttaa hänen kykynsä johtaa ketterässä ympäristössä.[34]

Taulukko 6: Churchmanin viisi tiedonhankintametodia

Leibniz	analysoija / rationalismi	tiedonhaku loogisen ja deduktiivisen päättelyn tuloksena. Matemaattisin keinoin ja uusi tieto kerätään vanhan tiedon päälle.
Locke	realisti / empirismi	tiedonhaku kokeilevan ja induktiivisen päättelyn tuloksena. Tiedon luomisessa vaaditaan konsensusta ja otetaan kaikkien mielipiteet huomioon.
Kant	idealisti / idealismi	tiedonhaku on synteettinen useiden tapojen yhdistelmä. Tietoa haetaan laajalti ja objektiivisesti ja hyväksytään vastakkaiset mielipiteet. Näkee muutoksen positiivisena.
Hegel	syntetisoija / dialektinen	tiedonhaku perustuu vastakkaisten ajatusten synnyttämään kehitykseen. Keskustelut ja väittelyt vastakkaisista mielipiteistä nähdään luovana prosessina.
Singer	pragmatisti / pragmatismi	tiedonhaku perustuu kaikkiin muihin malleihin ja tapoihin, joita ihminen käyttää ajattellessaan ja luodessaan tietoa.

[31]

Taulukko 7: Ketterien menetelmien johtajien kyvyt peilattuna heidän ajattelutapaan[31].

	Vuorovaikutustyyli	Innovointi / tapa tutkia	Näkee muutoksen	Tapa hankkia tietoa	Visionäärinen kyky
<b>Leibniz</b>	(-) kontrolloiva	(-) etsii itse vastaukset	(-) hidas muuttumaan	(-) etsii yhtä ja oikeaa vastausta	(-) faktoihin pohjautuva
<b>Locke</b>	(n) sovitteleva	(-) kategorisoi saatua tietoa	(n) ei vastusta muutosta; on mukana jos yksimielinen päätös	(-) etsii vastausta joka sopii tiimille	(-) faktoihin pohjautuva
<b>Kant</b>	(+) yhteistyöhaluinen	(+) laaja haku; muuttaa malleja vastaaviksi	(+) näkee muutoksen edistykseenä	(+) laaja tiedon etsiminen ja useat näkökulmat	(+) tulevaisuuteen suuntautunut tavoitteenasettaja
<b>Hegel</b>	(n) keskusteleva ja väittelee sisäisesti	(+) ajattelee yli normien ja rajojen	(+) Näkee konfliktin luovana prosessina ja tykkää eriävistä mielipiteistä	(+) etsii erilaisia tapoja katsoa asiaa	(n) spekuloiva, yhdistää ristiriitaisia näkemymiä mutta vaikeuksia päätöksenteossa
<b>Singer</b>	(n) ennalta-arvaamaton	(+) löytää uusia tapoja tehdä asioita	(+) hyväksyy muutoksen; joustava ja mukautumiskykyinen	(+) arvioi jatkuvasti uutta tietoa	(-) taktinen; toteuttaa askel kerrallaan
<b>(+) positiivinen (n) neutraali (-) negatiivinen</b>					

### 2.5.2 Käytännön ohjeita ketterään johtamiseen

Peter Deemer on tutkinut asioita, joita Scrum-tiimin johtajan (engl. ScrumMaster) olisi hyvä noudattaa. Taulukko 8 kuvaa asioita, joita johtajan tulisi toteuttaa johtaessaan ketterää Scrum-tiimiä. Taulukko 9 taas kuvaa asioita, joita Scrum-tiimin johtajan tulisi ehdottomasti välttää.

Taulukko 8: Scrum-tiimin johtajan lista asioista joita tulisi toteuttaa

Scrum-tiimin johtajan lista asioista, joita tulisi toteuttaa
Poista esteitä jotka tulevat Scrum-tiimin eteen
Auta tiimiä teknisten ongelmien kanssa
Järjestä säännöllisesti tapaamisia kasvokkain tiimin jäsenten kanssa
Opiskele tekniikoita, metodeita ja työkaluja joita tiimi hyödyntää
Suunnittele koulutusta ja taitojen kehittämistä tiimille
Seuraa toimialan uutisia ja uusimpia kehityssuuntia
Ennakoi tiimin tarvitsemia työkaluja, taitoja ja tarpeita
Hallitse budjettia sekä talouteen liittyviä odotuksia
Anna tietoa mitä ominaisuuksia ja toiminnallisuuksia tiimin pitää toteuttaa
Tee suorituskyykyarvioita ja anna palautetta tiimille
Pidä huoli tiimin jäsenien ura- ja kehityssuunnitelmista
Anna palautetta, kuinka jokin ominaisuus voitaisiin toteuttaa paremmin
Rekrytoi tiimiin sopivat jäsenet ja hanki tarvittaessa uusia osajia
Optimoi tiimin suorituskyykyä ja vaihda tarvittaessa henkilöt, jotka eivät sovi tiimiin

Taulukko 9: Scrum-tiimin johtajan lista asioista joita johtamisessa tulisi välttää

Scrum-tiimin johtajan lista asioista, joita johtamisessa tulisi välttää
Älä määrittele äläkä priorisoi tiimin työtehtäviä
Älä osoita yksittäisiä tehtäviä ja töitä tiimin jäsenille
Älä seuraa yksittäisten tehtävien suorittamista henkilötasolla, koska se on tiimityötä
Älä sotkeudu itse suorituvaiheeseen, koska on tärkeää, että tiimi tekee itse työnsä
Älä määrittele tiimin toteuttamisvauhtia, koska tiimi on itseohjautuva
Älä ota vastuuta tiimin suoriutumismuutoksista johdolle, koska tuoteomistaja vastaa tehtävien valmistumisesta ja aikataulusta isommassa kuvassa
Älä tuhlaa ylemmän johdon aikaa raporttoimalla projektin etenemistä, koska tuoteomistaja vastaa aikataulusta ja sen esittämisestä
Älä järjestä yhteisiä viikkotapaamisia, sillä Scrum-prosessi on omat päiväpalaverit ja aloituspalaverit, joissa tiimin jäsenet tapaavat.

### 3 Tutkimusaineisto ja -menetelmät

Tässä luvussa kuvataan käytetty tutkimusaineisto ja tutkimuksen metodologiset valinnat, sekä kerrotaan tutkimuksen toteutustapa ja käytetyt menetelmät. Luku on jaettu vielä neljään alalukuun, joista ensimmäisessä käydään läpi tarkemmin tutkimusmenetelmää ja sitä miksi on päädytty haastattelututkimukseen. Toisessa aliluvussa kuvaillaan haastattelujen suunnittelua sekä käydään läpi, miten haastattelujen tuloksia on analysoitu.

#### 3.1 Tutkimusmenetelmä ja sen perustelu

Tässä tutkimuksessa tutkimusmenetelmäksi on valittu haastattelu. Työssä keskitytään Soneraan ja sen palvelukehitysprosessiin, eikä aiheesta ole tehty kyseiselle yritykselle aikaisempaa tutkimusta. Tämän vuoksi voidaan todeta, että aihe on uusi tai ainakaan sitä ei ole kartoitettu tutkimusten avulla. Laadulliset menetelmät ovat hyödyllisiä, kun tutkitaan kartoittamattomia tai tutkimattomia kohteita [37][38]. Kvalitatiivinen tutkimus toimii hyvänä tiedonlähteenä, kun tietoa kohteesta etsitään syvälliseltä tasolta. Kvantitatiivisia tutkimuksia hyödynnetään teoriaosuudessa, mutta itse kohteena olevaa yritystä tutkitaan kvalitatiivisin menetelmin. Kun kvalitatiiviset menetelmät ovat antaneet riittävästi tietoa, voidaan perustellusti muodostaa myöhempää tutkimusta varten hypoteeseja. Jatkotutkimukset voivat olla kvantitatiivisia, jolloin palvelukehitysprosessia voidaan kuvata tilastollisin menetelmin ja saada lisätietoa tutkittavasta aiheesta.[39][38].

Tämän tutkimuksen tavoite ei ole päästä tilastolliseen yleistettävyyteen, vaan tarkoitus on tehdä mahdollisimman oikeita analyyseja tutkimusaineistoista. Tutkimuksen tavoite on herättää ajatuksia tutkimusalueesta ja antaa lisäarvoa yritykselle suorittaa kvantitatiivisia jatkotutkimuksia. Palvelukehitysprosessi ideoinnista markkinoille valmiiksi palveluksi on pitkä ja monimutkainen prosessi, jota on hyvin vaikea analysoida tuntematta taustalla piileviä syitä. Tämän vuoksi kvalitatiivisista tutkimusmenetelmistä valittiin haastattelu, koska se on paras menetelmä saada taustalla olevia syitä tutkittavasta kohteesta[39].

#### 3.2 Tutkimushaastatteluiden toteutus

Tämän tutkimuksen haastattelut ovat malliltaan puolistrukturoituja. Puolistrukturoidussa haastattelussa kysymyksillä ei ole ennalta määrättyjä vastausvaihtoehtoja, vaan haastateltavat voivat vastata haluamallaan tavalla [39][40]. Kysymyksien ei tarvitse olla tarkassa sanamuodossa, eikä niitä tarvitse esittää ennalta määrättyssä järjestyksessä. Tärkeintä haastattelumallissa on, että sen luonne on keskustelevalta, mutta haluttuihin kysymyksiin vastataan haastattelun aikana. [39]

Haastatteluja toteutettiin 14 kappaletta, joista jokainen pidettiin kasvotusten Soneran toimistotiloissa. Haastatteluja ei tallennettu, sillä sanatarkkojen sitaattien

sijaan oli olennaista kerätä mahdollisimman syvällistä tietoa. Haastatteluiden aikana vastaukset kirjattiin ylös ja osa keskusteluista jätettiin kirjaamatta, mikäli se ei liittynyt kysymyksiin tai aiheeseen. Haastattelujen luottamuksellisuuden vuoksi ja haastateltavien pyynnöstä nimiä ei mainita tässä tutkimuksessa. Haastateltavien rooli organisaatiossa mainitaan, koska sillä on merkitystä haastatteluiden kannalta eikä se tässä tutkimuksessa vaaranna anonymiteettiä.[39] Asema organisaatiossa on tämän tutkimuksen kannalta oleellinen asia, sillä eri asemassa oleville henkilöille on esitetty hieman toisistaan poikkeavia kysymyksiä.

Palvelukehitysprosessi on pitkä ketju ja funktionaalisessa organisaatiossa siihen liittyvien osastojen toimenkuvat eroavat sen verran paljon, että kysymyslistat vaihtelevat osaston mukaan. Funktionaaliset osastot ovat jaoteltu neljään kategoriaan taulukon 10 mukaan. Haastatteluja varten laadittiin kysymyslistat, joista esimerkkejä löytyy taulukosta 11. Kysymyslistat poikkeavat toisistaan haastateltavien funktionaalisen osaston mukaan.

Taulukko 10: Haastateltavat

Haastateltavat	Tehtävä	Yksikkö
4 kpl	Projektipäällikkö, kehittäjä, tuote-päällikkö, projektikonsultti	Linjaorganisaatiot / Projektijoh-taminen
4 kpl	Tuotealuevastaava, yksikön johtaja, osaston esimies	Tuotteet ja palvelut
4 kpl	Portfolion johtaja, yksikön johtaja, osaston esimies	Projektijohtamisen ja projekti-portfoliohallinta
2 kpl	Tarjoamavastaava, strategiajohtaja	Liiketoiminta ja tarjoama

Vaikka haastattelukysymykset poikkesivat hieman toisistaan, niissä oli paljon yhtäläisyyksiä. Kaikki haastattelut alkoivat kysymyksellä henkilön yleisestä roolista, sekä häntä pyydettiin kuvailemaan hänen roolinsa kautta palvelukehitystä. Tämän jälkeen siirryttiin ongelmakohtiin, jonka jälkeen päästiin pureutumaan yksittäisiin kysymyksiin, joiden tarkoitus oli saada tarkempaa tietoa syvällisemmistä pohdin-noista. Hyvänä haastattelutapana pidetäänkin tekniikkaa, jossa siirrytään laajoista helpoista kysymyksistä kohti yksityiskohtaisempia [39].

Taulukko 11: Esimerkkikysymyksiä haastateltaville

Kysymys
<b>Projektipäälliköt ja kehittäjät</b>
Kuvaile perus projektia
Mitä kehitysmalleja tai metodeita projekteissa on käytetty?
Mitkä ovat olleet ongelmakohtia kehittämisessä tai palvelun luomisessa?
Mitataanko toimintaa / kehitetäänkö mittaamista tai projektin sisäistä toimintaa systemaattisesti?
Mikä on ollut asiakkaan rooli palvelukehityksessä?
Mitä eroa ketterällä sekä lean-periaatteiden mukaisella ja perinteisellä kehityksellä?
Miten olet viihtynyt projekteissa, ketterä vs perinteinen?
Mitkä ovat nykyisen johtamismallin ongelmakohtia?
<b>Projektiportfolion hallinta</b>
Kuvaile projektinhallintaa projektiportfoliota
Onko ketterä ja lean-mallin kehittäminen tuttua?
Miten ketteryys ja lean-mallin vaatimukset näkyvät projektinhallinnassa?
Kuinka usein projekteja lopetetaan kesken kehityksen ja kannustetaanko tähän?
Mitkä ovat nykyiset johtamismallin ja portfoliohallinnan haasteet?
Miten kehityshankkeet priorisoidaan?
<b>Tuottet ja palvelut- ja liiketoimintayksikkö (* = Kysytty liiketoiminnalta)</b>
Kuvaile perus palvelukehityshanketta
Miten henkilöresurssit jaetaan projekteihin tai kehityshankkeisiin? (Miten ne tulisi jakaa?)
*Mitkä ovat olleet ongelmakohtia palvelukehityksessä?
Mitä tapahtuu kun tuote tai palvelu on valmis? Miten palvelukehitystä tulisi kehittää?
*Mitataanko toimintaa / kehitetäänkö mittaamista tai projektin sisäistä toimintaa systemaattisesti?
Mitkä ovat nykyisen johtamismallin ongelmakohtia?
Minkälainen johtamismalli halutaan tuotekehityksen tueksi?
*Kuinka usein projektia lopetetaan kesken kehityksen ja kannustetaanko tätä?
*Miten kehityshankkeet priorisoidaan?
*Miten liiketoimintapäätökset tehdään?
*Kuka päättää hankkeista?
*Mikä on asiakkaan rooli hankkeissa ja päätöksissä?

### 3.3 Haastatteluaineiston analysointi

Empiirinen aineisto esitellään seuraavassa luvussa, mutta näiden pohjalta tehtävä analyysi käydään läpi luvussa kuusi. Hyvän tutkimustavan mukaan analyysin ja tulosten erottaminen antaa mahdollisuuden kritisoida tehtyjä johtopäätöksiä ja analyysia kohtaan [39] [40]. Haastatteluita tehtiin yhteensä 14, ne olivat pituudeltaan noin tunnin mittaisia. Liiketoimintayksiköiden henkilöille pidetyt haastattelut olivat lyhyempiä, koska niiden tarkoitus oli validoida muissa haastatteluissa nousseita ole-



tuksia. Vaikka henkilöt toimivat tietyissä rooleissa he pystyivät vastaamaan muiden osastojen asioiden puolesta heidän tietonsa ja aiempien kokemusten vuoksi. Moni työntekijöistä on ollut useammassa osastossa töissä, jonka vuoksi ne pystyivät omaksumaan sekä liiketoiminta- että tuotevastaavan roolin.

Haastattelututkimuksen tarkoituksena on tehdä synteisiä aiempien tutkimusten ja empiiristen tulosten suhteen. Tutkimuksen validiteettia voidaan kasvattaa viittaamalla haastattelujen tuloksia aiempaan kirjallisuuteen. Tämä ei tarkoita sitä, että teorian ja löydösten tulisi tukea toisiaan, koska se voi jopa estää tutkimuksen tekemisen. Laadullisissa tutkimuksissa voidaan käyttää deduktiivista ja induktiivista päättelyä, sekä näiden välimallia.[39]

Tämä tutkimus hyödyntää induktiivisen ja deduktiivisen mallin välimuotoa. Valinta on tehty siksi, että aihetta ei ole vielä tutkittu kunnolla Soneralla ja siihen liittyy useita epävarmuustekijöitä. Aihetta on tutkittu useissa yrityksissä, mutta usein tutkimuksen kohteena on ollut ohjelmistokehitystä tekevä yhtiö. Myös näille yhtiöille suunnitellut ketterät ohjelmistokehitysmallit ovat usein toimivia sellaisenaan implementoituna, jolloin muutosten tuomia hyötyjä ei tarvitse pohtia ja perustella. Hypoteesien tekeminen etukäteen vaatisi kohteen tuntemista ja sen mitattavuuden ymmärtämistä [38]. Tämän tutkimuksen tarkoitus on löytää hyötyjä teoreettisista malleista käytännön kohdeyritykseen. Tästä syystä tutkimus tehdään induktiivisesti, joka tarkoittaa sitä, että teorialaue muodostuvat aineiston pohjalta.[39]

## 4 Tulokset

Tässä luvussa käydään ensin haastatteluiden tulokset läpi. Haastatteluiden tulokset on jaettu vielä useampaan alalukuun, jotta kokonaisuus olisi helpompi hahmottaa. Tämän jälkeen esitellään Soneran palvelukehitysprosessin nykytila.

### 4.1 Haastatteluiden tulokset

#### 4.1.1 Ongelmat palvelukehityksessä

Lähes jokaisessa haastattelussa hyvin pian yleisten ja laajojen kysymysten jälkeen siirryttiin palvelukehityksen ongelmien pariin. Kun kuvailevien vastauksien aikana haastateltava oli päässyt pohtimaan palvelukehitystä, ongelmista kysyttäessä saatiin paljon tietoa kerättyä. Ongelmiin liittyvien haastattelukysymysten tulokset löytyvät taulukosta [12](#)

Ongelmista kysyttäessä projektipäälliköt ja kehittäjät mainitsivat fasilitoinnin ja palaverien määrän muodostavan ongelmia. Isommassa mittakaavassa palveluiden kehittämistä hankaloitti kompleksiset IT-järjestelmät sekä resursointiongelmat. IT-järjestelmien hankaluus näkyy siinä, että järjestelmiä on monia ja niihin pitää integroitua. Järjestelmistä vastaavien henkilöiden tavoittaminen on hankalaa. Edellä mainittujen hankaluuksien lisäksi myös tietyn järjestelmän kehitysjonot saattavat olla niin pitkiä, että palveluun integroituminen voi viivästyttää projekteja useilla kuukausilla. Nämä seikat sekä resursointiongelmat aiheuttavat työn pirstaloitumista, joka myös erään haastateltavan mukaan aiheuttaa fokuksen katoamista, jos projekti venyy liian pitkäkestoiseksi. Yhtenä ongelmana nähtiin se, että teknologiayksikössä toimivilla henkilöillä ei ole projekteissa tuottovastuuta, vaan he vastaavat pelkästään tuotteen tappioista ja kustannuksista.

Tuotteet ja palvelut-yksikössä suurimpana ongelmana palvelukehityksessä nähtiin kompleksiset IT-järjestelmät. Myös projektien laajuus nousi useammassa haastattelussa esille ja tämän kerrottiin aiheuttavan hitautta palvelukehitysprosessissa. Kahdessa haastattelussa tuotiin esille osaamisen puute ja tässä viitattiin siihen, että tarvittavaa osaamista ei löydy tiettyihin projekteihin ja tämä luo ongelmia resursointiin ja projektien toteutukseen. Myös tuotteet ja palvelut-yksikössä nostettiin yhden henkilön toimesta ongelmaksi liiketoiminta- ja teknologiayksiköiden välinen kuilu.

Projektiportfoliohallinnan yksikön eli PPMO:n haastatteluissa huolenaiheina olivat projektien kesto, sekä niiden suunnitelmallisuuden puute. Yhteistä teemaa ei tästä yksiköstä löytynyt, mutta toisaalta haastateltavien roolitkin olivat toisistaan hieman poikkeavia. Liiketoimintalaskelma (engl. Business case) nähtiin ongelmalliseksi muuttaman haastateltavan osalta, koska nähtiin että sitä ei tehdä alusta alkaen yhdessä vaan se tehdään toteutusarvioinnin jälkeen. Liiketoimintalaskelmat olivat olleet myös kahden haastateltavan mukaan järjestelmällisesti ylioptimistisia.

Liiketoimintayksikössä ensimmäiseksi ongelmaksi tunnistettiin pullonkaulat, kuten projektipäälliköiden saatavuus ja henkilöresurssit. Järjestelmien kompleksisuus nousi myös yhteiseksi teemaksi. Yhtenä ongelmana nähtiin myös balanssin löytäminen uusien ja vanhojen kehitystapojen välillä. Yksi haastateltava mainitsi vielä tärkeän asian, joka tuottaa ongelmia siinä vaiheessa, kun palvelu tai tuote on saatu kehitettyä. *"Kun tuote on saatu markkinoille niin haaste on se, että meillä on niin laaja tuoteportfolio, että myynnin huomion ja kaistan saaminen on hankalaa. Yksi tuote saattaa hukkaa massaan".*

Taulukko 12: Ongelmat palvelukehityksessä

Tuotteet & Palvelut	Projektipäälliköt & Kehittäjät	PPMO	Liiketoiminta
<b>Kompleksiset IT-järjestelmät</b> Ihmiset vaihtuvat projektien välillä. Liian laajoja projekteja Hitaus. Speksaamisen kesto. Osaamisen vaje. <b>Resurssien puute.</b> Kuilu liiketoiminnan ja teknologian välillä	Projektin tavoitteissa ei varaa muutoksiin. Raportointi ja palaverit vievät liikaa aikaa. Asiakkaiden tapaamista hankaloitettu. Liian laajoja projektitilauksia. <b>Resursointiongelmat (kehityspotket tukossa).</b> Monet eivät tiedä mitä lean ja agile tarkoittaa. Allokaatioperusteinen tekeminen. Puuttuu yhteydet tekijöihin. <b>Kompleksiset IT-järjestelmät.</b> Työn pirstaloituminen. Teknologialla ei profittia, pelkkä loss.	Ei ole roadmappia isossa kuvassa -> Hankala toteuttaa strategiaa. Ensin ratkaisu, sitten mietitään business casea. Rahoja ei pystytty seuraamaan. Palvelukehitysketju on liian pitkä. Rahan saaminen kestää. Hinnoittelupäätös kestää. <b>Go-to-market ja kehitys pitäisi linkata yhteen.</b> Business caset ovat ylioptimistisia.	Balanssi uusien ja vanhojen tapojen välillä on vielä hakusessa. Ideoita ja ajatuksia on paljon enemmän kun kykyä toteuttaa. <b>Projektipäälliköiden saatavuus ja muut resurssit.</b> <b>Järjestelmien kompleksisuus aiheuttaa hitautta</b> organisaatioon. Laaja tuoteportfolio, myynnin huomion ja kaistan saaminen on hankalaa.

#### 4.1.2 Palvelukehityksen kehittäminen ja henkilöresurssit

Palvelukehityksen kehittämisestä kysyttiin sekä tuotteet ja palvelut-yksiköltä että projektipäälliköiltä ja kehittäjiltä. Ideana oli saada tietoa ongelmista ja haasteista kysyttäessä, miten he parantaisivat nykyistä prosessia. Tätä kysymystä kysyttiin tuotepäälliköiltä, tuotekehityspäälliköiltä ja heidän esimiehiltään ja osastopäälliköiltä. Parannusta nähtäisiin, jos saataisiin omia kehittäjiä, eikä heidän palveluitaan ostettaisi alihankintana. Nähtiin myös, että palvelukehitysketjun alussa ideoinnissa pitäisi hyödyntää asiakaspalvelukanavia ja vastaavia instansseja jotka ovat tekemisissä asiakkaiden kanssa. Suurin osa henkilöistä oli sitä mieltä, että tiimien pitäisi muodostua mahdollisimman laaja-alaisista osaajista. Tärkeänä asiana nähtiin myös yleisten suunnitelmakarttojen (engl. roadmap) luominen eikä liian yksityiskohtaisten osastojen suunnitelmien tekeminen. Tekijätasolla olevien henkilöiden haastatteluista nousi samoja asioita kehityskohteiksi. Tekijätasolla haastateltavat olivat projektipäälliköitä, kehittäjiä ja ratkaisuumistajia. Vastauksissa nousivat samat teemat, kuten eri osa-alueiden osaajien tuominen yhteen. Teknologian ja liiketoiminnan tuominen yhteen koettiin ehdottoman tärkeänä tehokkaan palvelukehityksen kannalta.

Monet haastateltavat mainitsivat ongelmissa jo ihmisten tavoitettavuuden olevan ongelma. Tämän lisäksi kysyttiin vielä suoraan, miten henkilöresurssit tulisi järjestää projekteissa, ja millä panostuksella ihmiset osallistuisivat projekteihin. Allokaatio-perusteinen osallistuminen projekteihin nähtiin haastavana ja lukuun ottamatta kahta henkilöä kaikki olivat sitä mieltä, että ainakin ydinryhmän tulisi työskennellä pelkästään yhdessä tiimissä. Haastateltavat puhuivat 100 prosentin osallistumisesta. Valtaosa haastateltavista tarkensi vielä, että ydinryhmän tulisi olla vain omassa projektissaan ja tarvittavia osajia tai sidosryhmiä ei koskisi samat vaatimukset. Vastauksissa tuli hieman vaihtelevuutta ydinryhmän koon suhteen, mutta pääosin puhuttiin 1-4 henkilön ryhmästä. Kaksi haastateltavaa oli sitä mieltä, että alle 50% osallistumista projekteihin ei pitäisi sallia, mikä tarkoittaisi sitä, että päähenkilöt ei saisi olla useammassa kuin kahdessa projektissa. Tuotteet ja palvelut-yksiköstä nähtiin tärkeänä, että ydintiimin täysi panostaminen luo motivaation menestymiseen. Tällöin henkilöt eivät voi piiloutua muiden tehtävien taakse. Samasta haastattelujoukosta ilmeni kuitenkin mielipide, että pienimmän kannattavan tuotteen rakennettuaan ryhmän ei tarvitsisi keskittyä enää palvelun kehittämiseen. Tekijätasolla nähtiin, että jokaisesta sidosryhmästä tulisi olla yksi edustaja täydellä panostuksella mukana, jotta asiat etenevät mahdollisimman nopeasti.

#### 4.1.3 Mittaaminen ja priorisointi

Ketterissä malleissa ja Lean-menetelmissä mittaaminen on hyvin olennainen osa palvelukehitystä ja prosessien kehittämistä. Tämän vuoksi haastateltavilta kysyttiin, miten projekteja ja palvelukehitystä on mitattu ja millä mittareilla. Tämän lisäksi monelle esitettiin jatkokysymys, miten mittaamista pitäisi suorittaa tai kehittää. Näiden haastatteluiden tulokset löytyvät taulukosta 13. Tekijätasolla haastateltavat sanoivat, että mittaaminen on hyvin vähäistä. Tavanomaisille projekteille on olemassa perinteiset seurannat, jossa katsotaan, miten rahaa on kulunut ja paljonko aikaa on käytetty. Tätä seurataan jälkikäteen eikä se ole aiheuttanut juuri mitään toimenpiteitä. Henkilöt kuitenkin sanoivat, että mittaamista pitäisi kehittää ja että palvelun käyttöastetta tulisi mitata, jotta palvelua voidaan muokata palautteen perusteella. Myös muissa yksiköissä ja haastatteluissa ilmeni, että vaikka mittaamista tehdään, niin mittareita ei seurata eikä niiden perusteella tehdä mitään muutoksia. Ketterien palveluiden kehittämistä varten tuli muutamia vastauksia, joissa pohdittiin kevyempiä mittareita tai indikaattoreita kuten asiakkaiden määrää, palvelun käyttöastetta, asiakkaiden tyytyväisyyttä, kehittäjien tyytyväisyyttä tai oppimista. Henkilöt, jotka pohtivat näitä mittareita, totesivat, että ylin mittari on kuitenkin palvelun tuottavuus. Tuotteet ja palvelut-yksikössä ilmeni, että perinteisten mittareiden käyttö on aiheuttanut hankaluuksia. Painottaminen pelkästään projektien määrään aiheutti laatuongelmia ja aikataulun mittaaminen aiheutti projektien pilkkoutumisen osaprojekteiksi. Osaprojektit koettiin hankalina, koska niiden päättyessä ei tullut mitään valmista lopputuotetta, jolloin projektilla ei ollut mitään tuottoa.

Soneralla joudutaan priorisoimaan paljon hankkeita, koska ideoita ja tekemisen arvoisia asioita on paljon, mutta henkilöresurssit eivät riitä kaikkien toteuttamiseen.

Taulukko 13: Palvelukehityksen mittaaminen

Tuotteet & Palvelut	Projektipäälliköt & Kehittäjät	PPMO	Liiketoiminta
<p>Projektit: Aikataulu – Budjetti Mittareita ei seurata. <b>Ei seurata miten ihmiset käyttää palveluita.</b> Jos mitataan projektien määriä -&gt; laatu tippuu. Loppukädessä yksinkertaista. Tuleeko liikevaihtoa ja asiakkaita. siirtyvätkö asiakkaat uuteen versioon. <b>Asiakkaiden määrän mittaaminen on helppo.</b> Yhteistyötä yksiköiden välillä mitattiin -&gt; Läpinäkyvyys kasvoi ja portfolio parani. SAFessa agile tiimien ja ohjelmistolisäysten arvon tuottoa suhteessa siihen aikaan missä niiden pitäisi tuottaa. Oppimista voitaisiin mitata.</p>	<p>KPI-mittarit yleensä huonoja. Pysyvyä aikataulussa, pysyminen ennusteissa, raha. Sisäinen kapitalisoitavan ja operatiivisen rahan jako ei toimi. Asiakastytyvyyttä vaikea mitata jos ei ole loppukäyttäjiä. Tulosta voidaan mitata. Pitäisi mitata: <b>Palvelun käyttöaste</b>, kanban- taululla lead timet. DevOps-tiimien suhde paljonko kehittämiseen ja paljonko operointiin. Pitäisi ymmärtää mihin rahaa käytetään.</p>	<p><b>Pitää siirtyä rahaprosessin mittaamisesta suoritteiden mittaamiseen.</b> -Suorittamisen nopeus (kun on tehtäviä) Projektimittarit: Raha, aikataulu, laajuus, rahavirta. Ennustamisen tarkkuus tärkeää pörssiyhtiölle Projektipäällikön pitäisi mitata suorituskkyä. PPMO mittaa liikevaihtoa / voittoa / hyötyä Kapitalisoitavan rahan käyttö suhteessa suunnitelmaan. Päätösten mittausta: Onko päätöksentekokriteerit tehty oikein, onko resurssit löytynyt? <b>Pitäisi olla leading indikaattoreita (esim. asiakasmäärä)</b></p>	<p>Kyllä mitataan. Ei mitata kovin syvällisesti. Projektien mittausta keskittyy että saadaan asioita valmiiksi tietyssä ajassa. Raha ei ole hyvä mittari jos ei ole vielä valmista tuotetta. <b>Uusia hankkeita voisi mitata asiakkaiden kautta: asiakastytyvyydellä ja käytettävyydellä.</b> Aika ja kustannukset ovat aina tärkeitä, mutta ne pitäisi olla suhteutettuna toteutettavaa tuotteeseen. Business Casejen toteutumista ei seurata tarkasti. Jälkikäteen ei tarkastella toteutuko business case kuinka hyvin. Kvantitatiivisia mittareita ei juurikaan ole.</p>

Palvelukehitysprosessin ymmärtämiseksi oli tärkeä kysyä, miten hankkeita priorisoidaan ja millä perusteilla. PPMO, joka vastaa projektiportfolioista ja rahankäytöstä, näki järjestelmällisesti priorisoinnin tapahtuvan johdon strategian mukaan. Ongelmana haastateltavat näki sen, että tuotekartat tai kehityskartat puuttuvat, minkä takia priorisointia ei voida tehdä kunnolla. Haastateltavat totesi, että liiketoiminta tekee kehityshankepäätöksiä, minkä vuoksi liiketoiminnalle pyritään antamaan mahdollisimman paljon tietoa. Liiketoimintapuolella nähtiin, että uudet liiketoimintalähdöt priorisoidaan kahdella kriteerillä. Ensimmäinen on strateginen sopivuus teleliiketoiminnalle ja toinen priorisointikeino on palvelun potentiaali. Ongelmana nähtiin liiketoimintalaskelmat (engl. Business Case), jotka ovat sinänsä objektiivisia ja sopivat priorisoitaviksi, mutta niiden tekijät eivät ole täysin objektiivisia. Liiketoiminnasta mainittiin myös foorumi, jossa kokoontuu useamman yksikön johtajia, joissa uudet hankkeet siirretään yhteiselle prioriteettilistalle ja näin se saa järjestyksensä tuotelistalla. Foorumin prioriteeteiksi mainittiin muutama peruste:

- Regulaatiomuutos. Lait
- Asiakaskriittiset toiminnallisuudet – sopimusvelvoitteet
- Liikevaihto – Kannattavuus – Liiketoimintalaskelmat
- Segmenttijaot. Kaikille segmenteille jotain tuotteita
- Asiakaskokemuksen vaikutus

Tuotteet ja palvelut-yksikössä priorisointia lähestyttiin hieman eri näkökulmasta. Eräs haastateltava mainitsi, että priorisointi tehdään täysin sattumanvaraisin perusteluin ilman mitään systemaattista keinoa. Vastauksessa käytettiin vertausta huutoäänestykseen ja todettiin että liiketoimintalaskelmat ovat useammin katteettomia lupauksia ja perustuvat ylioptimistisiin arvauksiin. Yksi haastateltava kuvaili, että priorisointi itsessään on ajankäytön haaskausta ja että priorisoinnin korostaminen ohjaa tekemistä väärään suuntaan. Useammassa vastauksessa todettiin, että kokeilussa olevassa SAFe-kehitysmallissa on painotettu priorisointityökalu jonka pitäisi vähentää priorisointiin kuluva aikaa ja tuoda prosessiin objektiivisuutta.

#### 4.1.4 Palvelukehityksen johtaminen ja projektien lopettaminen

Yhtenä ketterien menetelmien etuna on nopea aloitus ilman pitkää esiselvittelyjaksoa ja resurssien hankkimista. Tämä mahdollistaa projektien nopean tappamisen ilman pelkoa tuhlata ajasta ja rahasta. Haastateltavilta kysyttiinkin, lopetetaanko projekteja kesken ja kannustetaanko tähän. Haastattelutulokset tästä aiheesta löytyy taulukosta 14. Kaikki haastateltavat olivat järjestäen sitä mieltä, että kehityshankkeita lopetetaan liian harvoin kesken. Vain muutama haastateltavista pystyi muistamaan edes kaksi tapahtumaa joissa projekti olisi lopetettu kesken. Puolet haastateltavista koki, että projektien lopettamista kannustetaan, mutta puolet vastaajista näki, ettei siihen edes kannusteta. Tarkemmissa keskusteluissa syiksi paljastui epäonnistumisen leima, joka liitetään projektin lopettamiseen kesken kaiken. Lopettamisessa mukana olleilla henkilöillä oli pelkästään positiivisia kokemuksia projektien keskeyttämisestä. Liiketoimintayksiköstä mainittiin, että osa projekteista on kuitenkin semmoisia, että ne on pakko tehdä jolloin lopettaminen ei ole vaihtoehto. Suurin osa haastateltavista oli sitä mieltä, että nopeata keskeyttämistä (engl. fail fast) tulisi käyttää, mikäli palvelu ei ole riittävä hyvä. Tähän toivottiin myös systemaattista tapaa, jolla projekteja voidaan lopettaa.

Palvelukehityksen tehokkuuden kannalta hyvin oleellinen asia on, miten sitä johdetaan. Tätä teemaa lähestyttiin jälleen kysymällä nykyisen johtamismallin ongelmia ja tärkeimpiä kehityskohteita. Projektipäälliköiltä ja kehittäjiltä kysyttiin enemmän johtamismallin ongelmista ja miten palvelukehitystä tulisi johtaa. Kaikki haastateltavat tunnisti ongelmaksi tässä kentässä siiloutumisen. Siiloutumista on yritetty ratkaista foorumeilla, mutta niiden järjestäminen vie paljon aikaa erään haastateltavan mukaan. *"Pitäisi istua enemmän tuotekohtaisesti. Ideat syntyvät kahvitauoilla"*. Myös muut haastateltavat kokivat, että liiketoiminnan ja teknologian välillä on liikaa etäisyyttä ja osastot eivät kommunikoi keskenään riittävän tehokkaasti.

Liiketoiminnassa johtamisen haasteita nähtiin vain silloin kun projekti on niin laaja, että siihen osallistuu useampien liiketoimintayksikkö. Tällä tarkoitettiin kuluttaja- ja yritysasiakkaiden liiketoimintayksiköiden osallistumista samaan hankkeeseen, jolloin haastateltavan mukaan syntyy hankaluuksia vastuun ja kustannusten jakamisen kanssa.

Tuotteet ja palvelut-yksikössä suurimmaksi ongelmaksi todettiin se, että ihmiset

Taulukko 14: Palvelukehityksen johtamisen haasteita

Tuotteet & Palvelut	Projektipäälliköt & Kehittäjät	Liiketoiminta
<p><b>Teknologian ja liiketoiminnan pitäisi olla samassa tiimissä ja tehdä asioita yhdessä.</b></p> <p><b>Tuotto &amp; kulu-vastuu irtotetaan tosi kauaksi kehittämisestä.</b></p> <p>Infraa ja legacya pitäisi perinteisellä mallilla.</p> <p>OSS mallia ei kannata yrittää tehdä mielettömän ketterästi.</p> <p>El uskalleta antaa päätösvaltaa alas.</p> <p><b>Nykyisessä mallissa on liikaa tiedonjakotilaisuuksia, alle 5% kokouksista sisältää päätöksentekoa.</b></p> <p>Sijoittaja-ryhmän pitäisi johtaa kokonaisketjua niin kuin hallitus tekee.</p> <p>Tavotteiden tulisi olla yhteneväiset hallituksella ja tiimillä.</p> <p>Aikaraja, budjetti ja laajuus pitäisi olla sovittuna.</p>	<p><b>Siiloutuminen.</b></p> <p><b>Ratkaistu foorumeilla mutta ovat aikasyöppöjä.</b></p> <p><b>Pitäisi istua enemmän tuotekohtaisesti.</b></p> <p>Ideat syntyvät kahvitauoilla.</p> <p><b>Rooleja pitäisi sekoittaa eikä siilouttaa.</b></p> <p>Ketterät mallit eivät toimi jos on resurssiongelmia.</p> <p>Kilpailu resursseista -&gt; parhaat projektit saavat parhaat tekijät.</p> <p>Linkitys puuttuu tarjoaman ja IT alustojen sekä tuoteportfolioiden hallinnan ja tuoteratkaisuiden välillä.</p> <p><b>Siilot ovat liian korkeita ja kuulut liian syviä.</b></p> <p>Tehdään ylimääräistä eikä huomata tekemisen vaikutuksia jos tehdään tai jätetään jotain tekemättä.</p> <p><b>Tuotto &amp; kulu-vastuu puuttuu ratkaisuumistajalta</b></p>	<p>Liiketoimintaomistajan näkökulmasta käytännöt ovat hyvin kirjavia.</p> <p>Joissain on viikottain ohjausryhmiä ja joissakin on kerran puolessa vuodessa.</p> <p>Pitäisi olla joku tietty tapa miten projekteissa toimitaan.</p> <p>Päätöspisteet ovat samat kaikille.</p> <p>Projektipäälliköiden kanssa toimii yhteistyö.</p> <p>Hankalimpia projekteja on semmoiset missä on useampien liiketoimintayksiköiden kanssa toimintaa.</p> <p>Jos B2C ja B2B tekee projektia yhdessä. Tällöin vastuukysymykset jakautuvat ja kustannusten jakaminen tuottaa hankaluuksia.</p>

on jaettu teknologia- ja liiketoimintayksikköön. Useiden vastausten mukaan heidän pitäisi olla samassa tiimissä ja tehdä asioita yhdessä. Yksi haastateltavista totesikin, että voiton ja tappion vastuu on todella kaukana kehittämisestä. Yksi haastateltavista totesi, että perinteistä infrastruktuuri-rakentamista ei edes tulisi yrittää tehdä mielettömän ketterästi vaan pitäisi olla useampi eri ohjausmallit eri tyyppisille kehityshankkeille. Yhtenä johtamisen ongelmana mainittiin päätösvallan pysyminen liian ylhäällä, jolla tarkoitettiin sitä, että yhtiössä järjestetään paljon kokouksia, joissa pohditaan asioita, mutta ei tehdä päätöksiä. Haastateltavan mukaan päätösvalta pitäisi viedä toteuttavalle tiimille asti, jolloin ei tarvitsisi järjestää niin paljon turhia kokouksia, joita hän kutsui tiedonjakotilaisuuksiksi.

#### 4.1.5 Ketterien ja perinteisten menetelmien erot ja asiakkaan rooli

Osalle haastateltavista esitettiin vielä kysymys ketterien ja perinteisten menetelmien eroista. Kaikkien kanssa tähän kysymykseen ei riittänyt aika, varsinkin jos henkilö oli tehnyt kehitystä ketterillä menetelmillä kuten Lean startup- tai Scrum-mallilla. Taulukossa 15 näkyy haastattelun vastaukset menetelmien eroista. PPMO näki suurimman eron siinä, että ketterillä malleilla tuote saadaan nopeammin markkinoille. Arvon tuottamine tapahtuu nopeammin, koska Teliasoneran käyttämät TSPromo-kehitysmallissa on pitkä suunnittelujakso ja tuotetta määritellään loputtomiin ennen kuin sitä ruvetaan tekemään. Ketterän suurimpia hyötyjä nähtiin siinä, että asioihin

voidaan reagoida nopeammin ja että tekeminen on paljon läpinäkyvämpää. Eräs haastateltava mainitsi, että nykyisellä vesiputousmallilla voi olla tekevinään töitä pitkän aikaa eikä kukaan edes huomaa, ettei ole saanut mitään aikaiseksi.

Tuotteet ja palvelut-yksikössä nähtiin, että ketterään kehitysmalliin tulisi pyrkiä aina kuin mahdollista, mutta todettiin että joudutaan elämään sekä ketterässä että perinteisessä maailmassa samanaikaisesti. Tärkeänä seikkana ketterään siirtymisessä oli toimittajien hallinnointi ja heidän pakottaminen ketteriin menetelmiin. Vesiputousmallin suurimpana ongelmana nähtiin myös pitkät ja yksityiskohtaiset määrittelyjaksot.

Taulukko 15: Ketterien ja perinteisten menetelmien eroavaisuuksia

Tuotteet & Palvelut	Projektipäälliköt & Kehittäjät	PPMO
<p>Ketterään pitäisi pyrkiä aina kuin mahdollista.</p> <p>Joudutaan elämään molemmissa maailmoissa.</p> <p>Pääsemme ketterään kehitykseen jos voimme hallita toimittajia.</p> <p>Sidosryhmät kehittävät omalla aikataululla.</p> <p>Täydelliseen ketteryyteen on tämän vuoksi on hankala päästä</p> <p>Vesiputousmallin ongelma on speksaaminen.</p>	<p>Asiakas mukaan ideointi- ja lanseerausvaiheeseen.</p> <p>Palvelu ei ikinä täysin "valmis".</p> <p>Jatkuva kehittäminen ja mittaaminen. Vastataan nopeammin asiakkaan tarpeeseen.</p> <p>Vesiputouksessa ei varaa muutoksiin. Leanissa ongelmat nousevat nopeammin pöydälle ja niihin päästään käsiksi.</p>	<p>TSPromossa pitkä suunnittelujakso jossa speksataan loputtomiin.</p> <p>Ketterillä malleilla saadaan tuote nopeammin markkinoille.</p> <p>Tuotetaan arvoa nopeammin.</p> <p>Ketterillä malleilla rahaa vapautuu nopeammin (tämä on hyvä PPMO:lle)</p> <p>Vuoden päästä kaikki kehitysmallit ovat päivän päätyttyä samassa loppupisteessä.</p> <p>Agilen suurin hyötyä on että voidaan reagoida asioihin nopeammin. Suorittaminen on paljon konkreettisempaa.</p> <p>Waterfallissa voi surffailla pitkän aikaa eikä sitä välttämättä edes huomata jos on suoritusvajetta.</p>

Projektipäälliköt ja kehittäjät nostivat asiakkaan osallistamisen vahvasti mukaan ja sanoivat sen tuovan suurta eroa. Todettiin, että ketterällä kehityksellä tuote ei ole täysin valmis koskaan vaan se vaatii jatkuvaa kehittämistä. Ketterät ja asiakaslähtöiset mallit pystyvät vastaamaan nopeammin asiakkaan tarpeeseen. Eräs projektipäällikkö mainitsi, että vesiputous-mallilla tehtäessä ei ole varaa tai aikaa tehdä muutoksia, ellei niitä ole suunniteltu etukäteen, jota hyvin harvoin tapahtuu. Yksi kehittäjistä mainitsi että: *"Lean-mallilla ongelmat nousevat nopeammin pöydälle ja niihin päästään käsiksi.* Kehittäjiltä ja projektipäälliköiltä, jotka olivat osallistuneet Lean Service Creation-ohjelmaan, kysyttiin vielä kumman mallin mukaisessa kehitystyössä he viihtyvät paremmin. Lean Service Creation (LSC) oli koulutusohjelma, joka sisälsi Lean startup-mallin periaatteita ja palvelumuotoilua, joihin osallistui useampi henkilö Soneran organisaatiosta. Kaikki haastateltavat sanoivat, että ketterä



tekeminen on paljon innostavampaa. LSC projekteissa ihmiset viihtyivät paremmin, koska näissä sai tehdä tavallista enemmän päätöksiä itse. Yksi haastateltavista totesi asian selkeästi: *"Perinteisen mallin tarkoissa määritelmässä pysyminen aiheuttaa turhaa stressiä. Ehdottomasti ketterä"*.

Suurimpana tekijänä palvelukehityksen tehokkuudessa on lisäarvon luominen asiakkaalle. Tämän vuoksi jokaiselta haastateltavalta kysyttiin, mikä on asiakkaan rooli palvelukehityksessä. Jatkokysymyksenä useimmille esitettiin, miten asiakkaan rooli on muuttunut viimeisten vuosien aikana. Liiketoiminnalta kysyttäessä molemmat haastateltavat totesivat, että asiakkaan pitäisi olla kehityksessä keskeisessä roolissa. Haastatteluissa ilmeni, että parantamisen varaa olisi asiakkaan ottamisessa mukaan palvelukehitykseen. Kaikki haastateltavat totesivat, että asiakkaan rooli on kasvanut viimeisten vuosien aikana. Tarkemmin pohdittuaan eräs haastateltavista totesi: *"Kehitystyötä on tehty aiemmin yhteistyössä asiakkaiden kanssa, mutta nyt tekeminen on systemaattisempaa"*. Soneralla on hiljattain ruvettu käyttämään menetelmiä, joissa otetaan huomioon palvelumuotoilu ja yritetään tehdä palvelukehityksestä ketterää ja asiakaslähtöistä. Näissä kehitysmalleissa on olemassa tiettyjä harjoitteita, jotka ovat edesauttaneet asiakkaan osallistamisen palvelukehitykseen.

Kehittäjät ja projektipäälliköt taas totesivat, että asiakkaiden rooli on ollut järjestäen lähes olematon. Jossain projektissa oli luettu asiakastyytyväisyyskyselyitä yleisellä tasolla, mutta muuten hankkeen edetessä ei oltu tehty muita toimenpiteitä asiakkaan osallistamiseksi. Asiakkaiden kanssa kehittämistä tulisi kehittää, sillä viimeisissä hankkeissa oli ollut hankaluuksia löytää sopivaa tapaa toimia asiakkaiden kanssa. Ajankäytön ja asiakkaiden toiveiden toteuttamisen suhteen nähtiin vielä, että oppimista tarvitaan. Yksi projektipäälliköistä mainitsi, että asiakkaiden toiveet tulevat monien välikäsien kautta ja tästä tiedosta yritetään tehdä oikeita johtopäätöksiä. Haastateltavat mainitsivat myös esimerkkeinä monen miljoonan projekteja, joissa ei oltu keskusteltu asiakkaiden kanssa koko kehityshankkeen aikana. Kaikki totesivat kuitenkin, että viimeaikaiset muutokset palvelukehityksessä on tehty oikeaan suuntaan ja asiakasta ollaan on viimeisissä projekteissa. Yhtenä huolenaiheena nähtiin, että usein ei mietitä, miten jonkin asian toteuttaminen tai tekemättä jättäminen vaikuttaa asiakkaaseen.

## 4.2 Soneran palvelukehitysprosessi

Palvelukehitysprosessia tutkittaessa on oleellista tunnistaa nykyinen prosessi sekä tietää miten siinä toimitaan. Haastatteluissa nousi usein esille teema, että palvelukehitysprosessin alussa liiketoimintayksikkö tekee päätökset hankkeiden aloittamisesta. Liiketoiminnan rooli on hyvin keskeinen palvelukehitysprojektissa, minkä takia tämän yksikön edustajilta kysyttiin suoraan heidän rooliaan palvelukehityshankkeissa. Liiketoimintayksikön rooli on kerätä systemaattisesti markkinatietoa kehityksen pohjalle, seurata kilpailijoiden liikkeitä, asiakkaiden tarpeita ja alan teknologiatrendejä. Itse roolista haastateltava mainitsi selvästi: *"Meidän rooli on käynnistäminen ja liiketoiminta on tilaajan asemassa. Liiketoiminta tilaa esiselvityksen, jos on todettu,*

*että tällöinen palvelu tarvitaan". Samaa ilmeni myös muista haastatteluista, että liiketoimintayksikkö tekee päätökset ja heidän päätöstään kunnioitetaan myös muissa yksiköissä. Liiketoimintayksikön roolista vielä hankkeen käynnistyttyä henkilö jatkoi seuraavasti: "Kun projekti käynnistetään niin liiketoiminta toimii liiketoimintaomistajana. Kun palvelu on valmis, tehdään markkinoille siirtymisen päätökset ja toimenpiteet. Tähän kuuluu hinnoittelupäätökset, lanseerauspäätökset ja jalkauttaminen myyntiin."*

Liiketoiminnan edustajilta kysyttiin vielä tarkemmin liiketoimintapäätösten tekemisestä. Päätökset pohjautuvat Gartnerin raportteihin, johdon strategiaan ja asiakkailta kerättävään tietoon. Uusien kehityshankkeiden suhteen portfoliota hallitsee sitä varten perustettu tiimi, johon kuuluu segmentin strategiajohtajat ja yksikön johtajia. Perinteisempiä tuotekehityshankkeita varten on olemassa tietyt foorumit, joita varten on olemassa selkeä prosessi. Näillä foorumeilla ja ohjausryhmillä on tapana hankkeen konsensusta, mutta lopullinen päätösvalta on puheenjohtajilla.

Soneran teknologiayksikössä käytetty projektimalli mukailee vesiputousmallia tai portti-vaihe mallia. Kuten kuvasta 12 näkee, päätöspisteet (engl. Decision Point) ovat nimetty lyhykäisyydessään DP0 - DP6. Muutamille päätöspisteille on annettu eri nimiä, joista tärkeimmäksi nousevat ensimmäinen päätöspiste DP0, jota kutsutaan TG0:ksi. Kun löydetään toteuttamisen arvoinen projekti, tarkoitus on aloittaa esiselvitys, mutta ennen sitä määritellään esiselvityksen laajuus, budjetti ja resurssit. Tässä kohtaa tärkein tehtävä on löytää alustavat resurssit projektille, jotta päästään tekemään ensimmäistä ehdotusta investointifoorumille. DP0 on ensimmäinen päätöspiste, jossa projektin ehdotusta tarkastellaan ja tehdään päätös projektin toteuttamisesta. Päätös projektin toteuttamisesta tehdään investointifoorumissa.



Kuva 12: Soneran teknologiayksikön projektimalli

Hyväksytyn DP0 päätöksen jälkeen alkaa esiselvitysvaihe, jonka tarkoituksena on löytää paras tapa toteuttaa ratkaisu. Tässä vaiheessa etsitään vaihtoehtoisia ratkaisutapoja ja arvioidaan toteutettavan työn laajuutta. Tämän lisäksi tarkoitus on myös etsiä mahdollisimman paljon tietoa tuotealueesta ja ratkaisusta ja asiakkaiden ongelmista, jotka liittyvät suunniteltavaan ratkaisuun.

Esiselvityksen jälkeen seuraava päätöspiste on DP1, jossa tutkitaan projektin toteuttavuutta ja hyväksytään budjetti, aikataulu ja resurssit suunnitteluvaiheeseen. Tämä päätös tarkoittaa sitä, että projekti on linjassa yhtiön strategian ja liiketoimintatavoitteiden kanssa. Päätös hyväksymisestä tehdään projektien ohjausryhmässä. Kun päätös on hyväksytty, projekti siirtyy suunnitteluvaiheeseen. Suunnitteluvaiheessa ideana on määritellä yksityiskohtaisesti toteutettavaa tuotetta tai palvelua,

joka valittiin esiselvityksessä. Suunnittelun tarkoituksena on saavuttaa riittävän hyvä arvio projektin aikataulusta ja budjetista. Tähän vaiheeseen kuuluu myös vaatimusten määrittely, toimittajien ja kumppaneiden valinta, sekä liiketoimintamallin ja investointipäätöksen tarkentaminen. Tässä kohdassa valitaan myös kehitysmalli, joka voi olla ketterä tai vesiputousmallin mukainen.

DP2 tai TG2 on tärkeä päätöspiste projektin kannalta, jossa investointifoorumi hyväksyy projektin kustannukset ja sitoutuu projektin lopulliseen investointiin. Tämän päätöksen saaminen saattaa kestää 2-3 viikkoa riippuen foorumin aikatauluista. Kun DP2 on hyväksytty, projektin laajuus on lukittu eikä sitä voi muuttaa ilman erillistä muutospyyntöä. Hyväksytyn päätöksen jälkeen projektityöntekijät ryhtyvät hankkeen toteuttamiseen.

Seuraavaksi vuorossa on muotoilu tai suunnitelmien viimeisteleminen. Tähän vaiheeseen kuuluu ostotilausten tekeminen ja kaikkien komponenttien tarkempi suunnittelu. Projektipäällikön tulee päivittää viikoittain projektin status ja kuukausittain projektin ennuste. Tämän jälkeen tuleva päätöspiste on DP3, jossa projektin ohjausryhmä hyväksyy yksityiskohtaisen suunnitelman. Tämän jälkeen tarkkoihin suunnitelmiin ei voida tehdä muutoksia ilman hallittua muutosprosessia.

DP3 päätöspisteen jälkeen vuorossa on ratkaisun toteutus eli implementointi. Tässä kohtaa on tarkoitus toteuttaa kaikki aiemmissa kohdissa suunnitellut tuotokset. Hankitut tuotteet tai järjestelmät integroidaan olemassa oleviin järjestelmiin ja testataan toimivuus. Projektin hankkivan asiakkaan osallistamista tulisi hyödyntää myös toteuttamisvaiheessa. Toteutuksen jälkeen tulee päätöspiste DP4, jossa hyväksytään projektin suoritteiden sisältö ja laatu. Kun suoritteet on hyväksytty, voidaan aloittaa valmistelut tuotteen tai palvelun käyttöönottoon ja huolehtia projektin luovutuksesta kohdeyksikölle.

DP4 päätöspistettä seuraa tuotantoon vienti, joka tarkoittaa palvelun käyttöönottoa ja projektin luovutusta. Tuote ja palvelu viimeistellään ja testataan se tulevassa toimintaympäristössä ja vielä mahdollisen asiakkaan kanssa. Kun kaikki on suoritettu, tuote tai palvelu luovutetaan liiketoimintaomistajalle. Tätä seuraa päätöspiste DP5, jossa tarkistetaan, että toteutus vastaa alussa tehtyä vaatimusmäärittelyä. Tässä päätöspisteessä arvioidaan vielä testauksen tuloksia.

Kun DP5 on hyväksytty, siirrytään projektin lopettamisvaiheeseen. Projektin lopettamiseen kuuluu toimintojen sulkeminen, resurssien vapauttaminen, käyttöomaisuusinvestointien aktivointi, avoimien työnsitusten sulkeminen ja liiketoimintalaskelman päivittäminen. Lopettamisvaiheessa on myös tärkeää analysoida tietoja ja kerätä näkökulmia loppuraporttia varten. Projektin lopetusta seuraa viimeinen päätöspiste DP6, jonka jälkeen projekti on virallisesti lopetettu. Projektin tulisi täyttää seuraavat yleiset kriteerit:

- Projektin jäsenet ovat palanneet linjaorganisaatioihin
- Projektissa käytetyt välineet ja muut resurssit on palautettu
- Projektin dokumentit on arkistoitu

- Loppuraportti on laadittu ja esitetty projektiomistajalle
- Vastuuhenkilö projektinjälkeiselle arvioinnille on valittu ja hyväksytty projektiomistajan toimesta

Muutospyyntö tulisi aina jättää investointifoorumille, mikäli budjetti, aikataulu tai projektin laajuus muuttuu. Mikäli muutokset ovat pieniä ja kohdistuvat vain projektin laajuuteen tai aikatauluun, hyväksynnän voi tehdä projektin ohjausryhmä. Projektipäällikön ja projektin omistajan täytyy täyttää muutoslomake ja esittää muutokset ohjausryhmälle sekä liiketoimintaomistajan pitää lähettää muutospyyntölomake investointifoorumille. Myös projektin lopettaminen kesken vaatii muutospyyntölomakkeen täyttämisen eri foorumeille. Projektin lopettamisen tai valmistumisen jälkeen on vuorossa projektin arviointi. Projektin arvioinnin järjestäminen on projektin liiketoimintaomistajan vastuulla. Tämä tulisi järjestää 6 kuukautta projektin valmistumisen jälkeen.

Projektissa toimivilla henkilöillä on toisistaan poikkeavat roolit, joihin heidät valitaan linjaorganisaatiosta:

**Projektipäällikkö** on projektin toimitusjohtaja ja on vastuussa koko hankkeesta.

**Liiketoimintaomistaja** on vastuussa liiketoimintapäätöksistä ja edustaa projektia investointifoorumilla sekä toimii projektin johtoryhmässä.

**Ratkaisuomistaja** on vastuussa projektin teknisestä ratkaisusta, päätöksistä ja on velvoitettu ilmoittamaan ohjausryhmälle päätöksistä.

**Projektiomistaja** vastaa hankkeen käytännöistä, toimituksista ja suoritteista. (rahat, resurssit, aikataulu ja laatu). Toimii projektin johtoryhmässä.

**Resurssiomistaja** on linjaesimies, joka vastaa resurssien allokoinneista projektin tarpeisiin, sekä vastaa myös lopputuotteen laadusta.

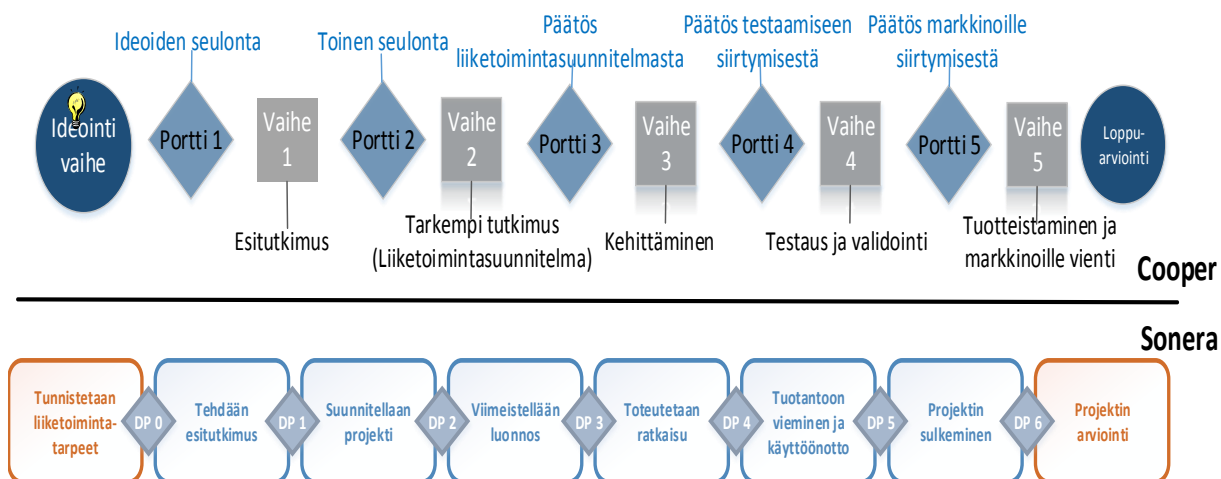
## 5 Johtopäätökset ja analyysi

Tämä luku käsittelee edellisen luvun empiiristen tulosten analysointia. Saaduista tuloksista etsitään yhtäläisyyksiä ja vastakkaisia näkökulmia ja näitä tuloksia peilataan teoriaan ja aiempiin tutkimuksiin. Tämä luku jakautuu tulosten analysointiin, kehitysehdotuksiin, johtopäätöksiin ja mahdollisten jatkotutkimusten aiheisiin.

### 5.1 Tulosten analysointi

#### 5.1.1 Projektikehitysmalli

Soneran teknologiayksikön projektikehitysmalli mukailee hyvin tarkasti Cooperin vaihe-portti-mallia[17]. Kuvassa 13 on kuvattu sekä Cooperin malli, että Soneran malli allekkain. Tästä huolimatta kahden ensimmäisen päätöspisteen (DP0 ja DP1) jälkeen suunnitteluvaiheessa projekti valitsee kehitysmallin, joka voi olla ketterä tai vaihe-portti-mallia mukainen. Tämä on ristiriidassa sen asian kanssa, että seuraavan päätöspisteen jälkeen (DP2) ei voida tehdä muutoksia projektiin ilman raskasta muutosanomusprosessia. Agile manifeston julkaisun mukaan muutoksiin tulisi vastata ja odottaa niitä ennemmin kuin yrittää pysyä alkuperäisessä suunnitelmassa. Myös asiakkaan vahva osallistaminen vasta toteutusvaiheessa on kyseenalaista, koska muutoksia ei voi enää tehdä ja vaatimusmäärittely on tehty suunnitteluvaiheessa tai viimeistään suunnitelmien viimeistelyvaiheessa. Eräs projektipäällikkö totesi kuitenkin, että projektikehitysmallin orjallinen noudattaminen ei ole välttämätöntä ja mainitsikin toiminnan olevan ketterämpää kuin tarkka vaiheiden ja porttien toteuttaminen. Monet haastateltavat puhuivat ketteristä menetelmistä yleisellä tasolla, mutta eivät nimenneet näitä. Ketteristä menetelmistä puhuttiin projektipäällikköiden yksikössä työkaluna, joita projekti voi käyttää toteutusvaiheessa.



Kuva 13: Soneran teknologiayksikön projektimalli ja Cooperin vaihe-portti-malli

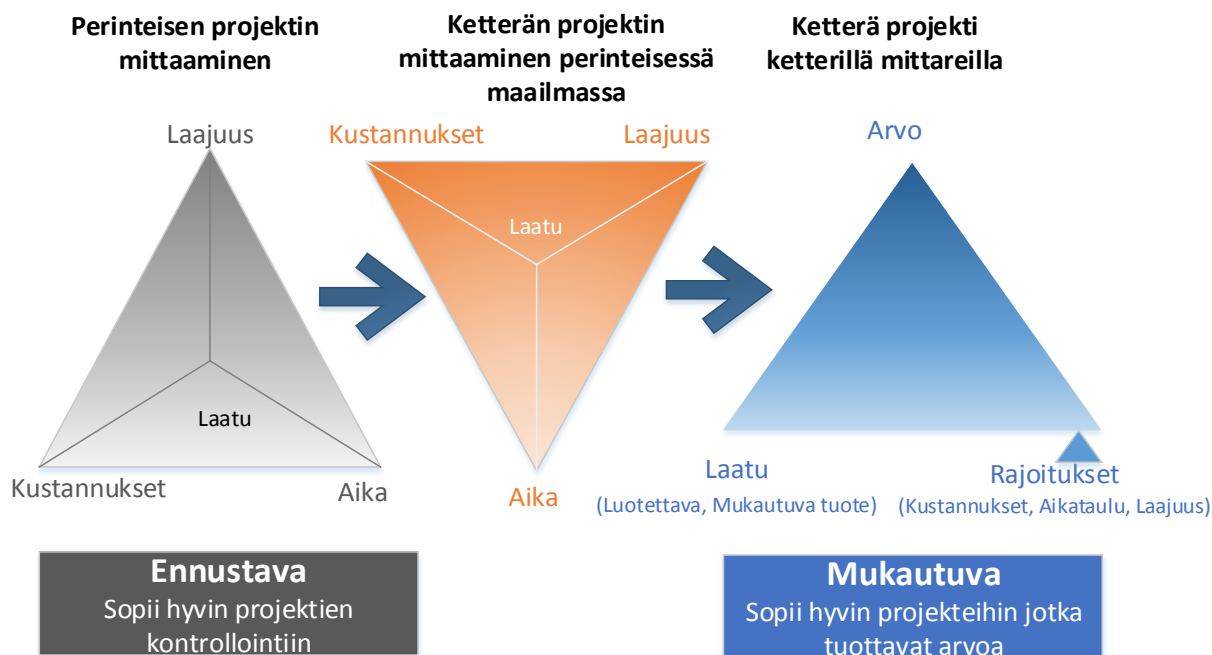
Soneralla on toteutettu ketterää kehittämistä, mutta se vaatii projektikehitysmallin TSPromo:n muokkaamista niin, että osa porteista jätetään tekemättä ja ketterille tiimeille tehdään raamipäätös ensimmäisessä päätöspisteessä (TG0), jonka jälkeen ne eivät toteuta perinteistä mallia. Ketterissä projekteissa tai kehitysohjelmissa hankkeilla on kyllä päätöspisteitä, mutta ei niin tiheästi eikä niitä ole yhtä useita. Projektiportfolion hallinnasta mainittiin toteutusmalleja, joissa rahaa luovutetaan ketterälle tiimille kvartaaleittain sitä mukaan, kun ryhmä on toteuttanut tehtäviä ja suoritteita, sekä saanut aikaan valmiiksi niitä osia, joita se on luvannut tehdä.

### 5.1.2 Hankkeiden laajuus ja mittaaminen

Projektien laajuus (engl. scope) nähtiin useissa hankkeissa olevan liian suuri tai epäonnistuvan pysymään muuttumattomana. Kehittäjien ja projektipäälliköiden mukaan johtajat tilaavat liian laajoja projektitilauksia ja usein on epäselvää vielä tarkalla tasolla, mitä näillä tilauksilla ja projekteilla yritetään saavuttaa. Myös määrittelyn keston nähtiin olevan ongelma palvelukehitysprosessissa. Kuvan ?? mukaan voidaan nähdä, että suurten ja keskisuurten projektien onnistuvuus vesiputousmallilla on hyvin pieni[24]. Ketterässä kehittämisessä aikataulun, budjetin ja laajuuden ylittämistä ei nähdä niin vaarallisena asiana, koska arvon tuottaminen asiakkaalle nähdään tärkeämpänä. Edellä mainittujen rajoitusten muuttuminen ei ole paha asia, mikäli asiakkaan kokema arvo kasvaa samassa tai suuremmassa suhteessa kuin budjetti ja käytetty aikataulu.[41]

Kuvassa 14 nähdään kolme kolmioita, joiden mukaan projekteja voidaan mitata. Ensimmäinen kolmio kuvaa perinteisten kehitysmallien tyypillisimpiä mittareita. Vaihe-portti- ja vesiputousmallin kehitystä mitataan usein käytetyllä rahalla, ajalla ja laajuudella. Perinteiset projektit on määritelty etukäteen, jolloin alkuperäisten suunnitelmien muuttuessa raha ja aika muuttuvat projektin edetessä. Toisessa kolmiossa kuvataan ketterän projektin mittaamista perinteisillä mittareilla. Tässä mallissa projektin onnistuvuutta mitataan käytetyn ajan, rahan ja laajuuden perusteella. Oikeanpuolimmainen kolmio taas kuvaa ketterää kehitysprojektia ketterin menetelmin. Tässä mallissa hyödynnetään edelleen perinteisiä mittareita, mutta ne eivät ole projektin tavoite. Muita mitattavia asioita ovat arvon luominen asiakkaalle ja laatu. Tässä mallissa lisäarvon luominen saattaa vaatia muutoksia projektin budjettiin ja aikatauluun, mutta tällöin saavutetaan parempi laatu ja suurempi arvo asiakkaalle.[41]

Palvelukehityksen ja projektien mittaamisesta kysyttiin kaikilta yksiköiltä ja vastaukset tekijätasolla olivat lähes yksiselitteisesti, että mittaaminen on hyvin vähäistä. Mittarit ovat olleet juuri perinteiset vesiputousmallin mukaisia, eli paljonko rahaa ja aikaa projektiin on käytetty. Muiden yksiköiden mukaan mittaamista kyllä tehdään, mutta se ei ole kovin syvällistä. Portfoliohallinnassa todettiin, että pitäisi siirtyä rahaproessin mittaamisesta suoritteiden mittaamiseen. Kaikilla yksiköillä oli kyllä hyviä näkemyksiä siitä, mitä tulevaisuudessa voisi mitata, mutta systemaattinen lähestymistapa puuttui. Yhdeksi syyksi tähän nähtiin se, että projektien omista-



Kuva 14: Projektihallinnan mittaaminen perinteisellä ja ketterällä mallilla

juus, kehittäminen ja johtaminen on hajautettu niin laajalle organisaatioon, että vastuu jakautuu myös mittaamisesta. Osa mittareista palvelee projektiportfoliota ja osa palvelee tuotepäällikköä, mutta tehokkaiden mittareiden luominen kehitysvaiheessa vaatisi kaikkien yksiköiden vahvempaa yhteistyötä. Hyviksi mittareiksi mainittiin välitavoitteita, kuten pienimmän kannattavan tuotteen luomisvaiheessa saatujen testiasiakkaiden määrällä ja palvelun käyttöasteella. Liiketoiminnassa oli paras yleisnäkemyks siitä, mitä palveluiden ja palvelukehityksen mittaamisella pitäisi saavuttaa.

### 5.1.3 Yleisimmät haasteet palvelukehityksessä ja tiimien johtaminen

Suurin yksittäinen syy palvelukehityksen ongelmallisuuteen nähtiin kompleksisissa IT-järjestelmissä. Tämä on ominaista operaattoriliiketoiminnassa ja vakuutusliiketoiminnassa, jos yrityksellä on historian ja regulaation painolasti kannettavanaan.[42] Koska yhtiöiden tietojärjestelmät ovat massiivisia, toimittajat ja kumppanit ovat myös suuria Tiedon ja Accenturen kaltaisia. Vanhat ja massiiviset tietojärjestelmät ovat useissa kehityshankkeissa pullonkauloja ja niihin integroituminen vaatii aikaa, sillä usein niiden järjestelmien kehitysjonot ovat pitkiä.

Yhtenä merkittävämmistä ongelmista niin palvelukehityksessä kuin sen johtamisessa nähtiin toteuttavan teknologia- ja liiketoimintayksikön välinen kuilu. Useampi haastateltava puhui siiloista tai kuiluista, joskin yhteinen nimittäjä oli päivittäisen yhteistyön puute. Soneralla on projekteja, joissa liiketoiminta ja teknologia yhdistyvät, mutta useissa tapauksissa näiden sovittaminen on iso haaste. Eikä ole montaa

hanketta tai tuotetta, joissa molempien yksiköiden edustajat istuisivat vierekkäin. Haastatteluissa ilmeni ongelmaksi juuri se, että toteuttavalla teknologiayksiköllä ei ole tulostavasta projektista. Vastuu toteutusvaiheessa on usein kulujen ja aikataulun hallitsemisessa eikä muissa tekijöissä. Suurimmassa osassa haastatteluissa toivottiin, että kehityshankkeen ydintiimin tulisi osallistua vain ja ainoastaan yhteen projektiin kerrallaan.

Edellä mainitut seikat toimivat hyvin silloin, kun tiimit järjestetään ketterien ja Lean Startup-menetelmien mukaan. Tällöin tiimit ovat itsenäisiä, kyvykkäitä ja pystyvät tekemään kaikki tarvittavat ratkaisut ja hoitavat päätöksenteon itse. Tällaisessa tiimissä on pakko olla mukana henkilöitä useammasta yksiköstä, jotta tekeminen ei keskeydy, kun kysymyksiä herää eri osa-alueista. Tiimien luominen tuotealueittain helpottaa myös resurssien läpinäkyvyyttä, kun ihmisten aikaansaannoksia mitataan visuaalisin keinoin. Tällöin henkilöt eivät ole allokointiprosenttien takana piilossa, eivätkä he voi olla käytettävissä muutamia tunteja viikossa tiettyihin projekteihin. Joissakin haastatteluissa mainittiin juuri hyvin siitä, että täyden vastuun ottaminen projektista on tärkeä asia ja läpinäkyvyys paljastaa eri osa-alueiden suoritusvajeet. Tuotteet ja palvelut-yksikössä mainittiin hankaloittaviksi seikoiksi myös resurssivajeet ja osaamisen puute, joihin on helpompi tarttua, mitä läpinäkyvämpää toiminta on.

Tiimien muodostamista varten on tärkeää, että ketterien tiimien eri roolit täytetään. Näitä rooleja on useita, mutta kuten Agile manifeston julkaisussa mainitaan, yhteistyö on tärkeämpää kuin tarkka dokumentaatio. Tämä asettaa tiettyjä vaatimuksia hyvin toimivien tiimien henkilöille. Myös tiimien johtajien tulisi olla persoonaltaan ja johtamistyyppiltään sopivia henkilöitä ketterän ja asiakaslähtöisen palvelukehityksen kannalta. Kuten taulukosta 16 nähdään, kontrolloivan esimiehen ja rationaalisen ajattelijan soveltuvan huonoiten ketterään kehitysmalliin. Hiljaisen tiedon hyödynnettävyys kasvaa, kun kaikkea ei tarvitse määritellä etukäteen ja dokumentoida vaan asioista keskustellaan ja ajatuksia vaihdetaan tiimin jäsenten kesken. Monialaisten osajien tiimiä muodostettaessa ketterään kehittämiseen sopii parhaiten Kantilainen johtaja, joka on idealistinen ja pyrkii tutkimaan mahdollisimman laaja-alaisesti ongelmakenttää. Tähän johtamistyyliin kuuluu vastakkaisten näkökulmien hyväksyminen ja ennen kaikkea muutoksen näkeminen positiivisena tekijänä. Ketterissä menetelmissä henkilöiden yhteistyön korostaminen ja sen kehittäminen on erityisen tärkeää, jota myös esimiehen tulisi vaalia tiimejä johtaessa ja muodostaessa.

#### 5.1.4 Ketterien menetelmien tuomat muutokset ja asiakkaan rooli

Siirryttäessä perinteisestä kehittämisestä ketterään kehitykseen palvelukehityksessä muuttuvat innovaatiokäytänteet. Näitä muutoksia on tutkittu aiemmissa tutkimuksissa ja merkittävämät erot ovat tiivistetty hyvin taulukossa 17. Suurin osa ongelmista, joita Soneran palvelukehityksessä on ilmennyt haastattelujen mukaan, on ollut suoraan verrannollisia perinteisten palvelukehitysmallien ongelmiin.[43] Kevätsalon tutkimuksen mukaan siirryttäessä ketteriin menetelmiin suurin osa näistä ongelmista poistuu. Suurin yksittäinen tekijä muutoksessa on päätöksenteko ja sen



	Vuorovaikutustyyli	Innovointi / tapa tutkia	Näkee muutoksen	Tapa hankkia tietoa	Visionäärinen kyky
<b>Leibniz</b>	(-) kontrolloiva	(-) etsii itse vastaukset	(-) hidas muuttumaan	(-) etsii yhtä ja oikeaa vastausta	(-) faktoihin pohjautuva
<b>Kant</b>	(+) yhteistyöhaluinen	(+) laaja haku; muuttaa malleja vastaaviksi	(+) näkee muutoksen edistysenä	(+) laaja tiedon etsiminen ja useat näkökulmat	(+) tulevaisuuteen suuntautunut tavoitteenasettaja
<b>Leibniz</b>	Rationalistinen ongelmanratkaisu tapahtuu loogisen ja deduktiivisen päättelyn tuloksena. Lähestyy ratkaisua matemaattisin keinoin ja kerää uutta tietoa vanhan tiedon päälle.				
<b>Kant</b>	Idealistinen ongelmanratkaisu tapahtuu synteettisesti useiden tapojen yhdistelmänä. Hakee tietoa laajalti ja objektiivisesti ja hyväksyt vastakkaiset mielipiteet. Näkee muutoksen positiivisena.				

Taulukko 16: Johtajien persoonien vastakohdat ketterään kehittämiseen.

nopeutuminen. Kun ryhmän aikajaksot ovat kahden viikon mittaisia, ryhmällä ei ole aikaa odottaa virallisia foorumeita ja päätöskokouksia, joihin kiireellisten johtajien läsnäoloa vaaditaan.[43]

Taulukko 17: Innovaatiokäytänteiden muutos siirryttäessä perinteisistä kehitysmalleista ketteriin[43]

Muuttuva käytäntö	Muutos
Päätöksenteko	<ul style="list-style-type: none"> <li>Päätöspisteistä jatkuvaan päätöksentekoon</li> <li>Päätöksentekomahdollisuudet eri yksiköiltä kehitystiimille</li> </ul>
Yhteistyö ja kehittäjien roolit	<ul style="list-style-type: none"> <li>Projektikokouksista epävirallisiin verkostoihin</li> <li>Kehittäjät muuttuvat konsulteista ja toteuttajista tasavertaisiksi suunnittelijoiksi</li> </ul>
Aikataulu ja resurssit	<ul style="list-style-type: none"> <li>Nopea aikataulu ja nopea eteneminen</li> <li>Henkilöresurssit osa-aikaisista kokoaikaisiksi</li> <li>Välineet ja tilat joustavammiksi</li> </ul>
Suunnitteleminen	<ul style="list-style-type: none"> <li>Esisuunnittelusta jatkuvaan suunnitteluun</li> </ul>

Palvelukehitysprosessin kehittämisestä haastatteluissa saatiin aikaan useita ja mielenkiintoisia keskusteluja. Itsenäisten ja kyvykkäiden tiimien muodostamista varten tarvittaisiin laaja-alaisia osajia, jotka pystyvät toteuttamaan monenlaisia suoritteita. Omien kehittäjien hankinta nostettiin monissa keskusteluissa esiin, sillä tämä vähen-

täisi määrittelyä, jota tarvitaan, kun työtä tilataan toimittajilta. Monialaiset (engl. Cross-functional) tiimit nähtiin tärkeänä palvelukehityksen tehostamiselle. Tämä on edellytyksenä ketterille ja lean-startup-tiimeille, jotta vastuu voidaan antaa kehittäjätiimille. Useammassa haastattelussa tuotiin myös esille asiakaspalvelukanavien ja muiden vastaavien instanssien hyödyntämistä palvelukehityksessä ja ideoinnissa. Ketterän kehittämisen perusmenetelmiin kuuluu asiakkaan vahva osallistaminen, joten näissä kanavissa olisi helppoa lähestyä asiakkaita ongelmahaastattelujen tai mielipiteiden keräämisen suhteen. Myös asiakkaiden kanssa useammin tekemisissä olevat henkilöt voivat tuoda tärkeää osaamista asiakkaiden lähestymisestä sekä heidän tarpeidensa analysoinnista. Eric Riesin Lean Startup-mallissa juuri nopea testaaminen asiakkailla ja asiakkaan ongelmien kartoittaminen ja ratkaiseminen on tärkeää luotaessa palveluita asiakkaille.

Asiakkaan rooli on ollut vaihteleva Soneran toteuttamissa kehityshankkeissa. On tehty isoja projekteja, joissa asiakkailta ei ole kysytty mitään ja toisaalta taas vuosikymmen aikaisemmin on tehty jotain projekteja yhteistyössä asiakkaiden kanssa. Keskimäärin asiakkaan rooli on kuitenkin ollut hyvin pieni ja lähinnä asiakastytyväisyyskyselyitä on luettu projektin aikana. Perinteisissä projekteissa asiakkaiden tarpeet ovat tulleet monista eri lähteistä ja useiden välikäsien kautta. Tässä on ollut havaittavissa muutosta sekä havaintojen että haastatteluiden perusteella. Asiakaslähtöinen kehittäminen vaatii opettelua, sillä eräänkin haastattelun mukaan liiallinen asiakkaan toiveiden toteuttaminen aiheuttaa ongelmia. Tähän mikä tahansa oppivan organisaation työkalu tai oppimisprosessin liittäminen auttaa, sillä tällä alueella systemaattista tekemistä ja oppimista tarvitaan. Lean startup-mallissa palvelukehitys lähtee rakentamisesta, mittaamisesta ja oppimisesta, mutta toiminnan kehittämisen ominaisuuksia on myös Scrum-kehitysmallissa ja muissa ketterissä menetelmissä.

### 5.1.5 Hankkeiden lopettaminen ja priorisointi

Kaikilta haastateltavilta kysyttiin projektien tappamisesta tai niiden lopettamisesta kesken kehityksen. Kaikki haastateltavat vastasivat, että hankkeita lopetetaan liian harvoin kesken ja vain muutamia esimerkkitapauksia nousi esille viimeisen kahden vuoden aikana. Puolet haastateltavista koki, että projektien lopettamiseen kannustetaan, mutta toinen puoli koki, ettei tähän edes kannusteta. Monet mainitsivat projektien lopettamiseen liittyvän epäonnistumisen pelko ja leimautuminen siitä, että on ollut tällaisen projektin johtaja tai omistaja. Nopeasta lopettamisesta (engl. fail fast) oltiin positiivisesti innostuneita, mutta työkaluja tähän malliin ei löydy. Soneran nykyinen projektijohtamismalli on sen verran kankea, että kun aloitus on niin hidasta, niin lopettaminen on aina raskas ja vaikea päätös. Soneran teknologiayksikön projektikehitysmallissa kuluu useampi kuukausi siihen, että tehdään esitutkimusta, suunnittelua ja saadaan henkilöresurssit kerättyä kasaan. Lopettamis päätöksen jälkeen kuluu aina useampi kuukausi ennen kuin päästään uuden aiheen pariin ja henkilöt ovat suorittaneet uuden aiheen esitutkimusta ja suunnittelua.

Myös priorisoinnin tekemisessä ja projektien lopettamisessa ja aloittamisessa nähtiin

yhtäläisyyksiä. Nykyisessä mallissa kehityskarttojen (engl. roadmap) puuttuminen nähtiin haasteena, kun kehityshankkeita pohditaan yksikkörajojen yli. Johdon strategian toteuttaminen priorisointimielessä nähtiin yksiselitteisenä ja sen jalkautuminen operatiiviselle tasolle ei vaikuttanut olevan haastavaa. Ainoat ongelmat priorisointien suhteen syntyväti niiden priorisoinnin objektiivisuudesta ja sen systemaattisesta mitattavuudesta. Liiketoiminnan priorisointi oli hyvin perusteltua, mutta kaikista yksiköistä todettiin, että liiketoimintalaskelmien mukaan priorisoiminen ei ole yksiselitteistä. Liiketoimintasuunnitelmien perusteella tehtävä priorisointi johtaa kilpailuun, jossa liiketoimintalaskelmista tehdään ylioptimistisia. Tuotteet ja palvelut-yksikössä on yritetty vähentää priorisointiin käytettävää aikaa ja keskitytty löytämään resursseja tarvittaviin projekteihin. Projektiportfolion hallinnassa oli hyvä idea toteuttaa hankkeita ketterästi kvartaali kerrallaan ja seurata kehitystiimin toimintaa heidän luomansa arvon kautta. Mikäli arvo ei kasva riittävän tehokkaasti suhteessa käytettyihin rajoitteisiin kuten aika ja raha, projektille ei anneta rahaa ja se lopetetaan. Skaalaamistyökaluissa on nähty priorisointiin auttavia työkaluja, kuten SAFe-mallissa oleva painotettu priorisointityökalu (engl. Weighted Shortest Job First). Ketterissä malleissa mitataan asiakkaan kokemaa arvoa eikä palvelun tuottamaa rahaa yritykselle. Tässä kohtaa on tärkeä muistaa, että vaikka yrityksen tärkein tavoite on tuottaa arvoa omistajille, palkan maksavat asiakkaat eikä yritys.

## 5.2 Kehitysmallien hyödyt Soneran palvelukehityksessä

Lean Startup-kehitysmallin ja ketterien mallien kuten Scrum-mallin käyttöönotto Soneran palvelukehityksessä hyödyttäisi vastaamalla useaan haasteeseen, joita nykyisessä systeemissä on. Taulukossa 18 on tuotu esiin mitä Soneran palvelukehityksen haasteita Lean startup-menetelmät torjuvat. Taulukossa 19 on taas tuotu esiin, miten Scrum lieventää nykyisen palvelukehityksen haasteita. Kompleksisiin IT-järjestelmiin ei voida vaikuttaa yksinomaan omalla kehitysmallilla, mutta toimittajilta voidaan vaatia läpinäkyvämpää tekemistä ketterien mallien avulla, jolloin toimitusvälejä voidaan lyhentää. Määrittelyjaksojen ja suunnittelujaksojen keskimääräistä kestoajaa voidaan lyhentää Lean Startup-menetelmien avulla. Tälle kehitysmallille on olennaista nopea tutkimusjakso, jossa pyritään löytämään viimeisin tieto ja tehdään sen perusteella hypoteeseja. Hypoteesien jälkeen niitä lähdetään testaamaan asiakkailla ja yritetään haastatella asiakkaita oikeiden ongelmien löytämiseksi.

Asiakkaan vahva osallistaminen projektiin alusta alkaen eikä vain määrittelyjaksolla takaa sen, että jokainen päätös luodaan asiakkaan arvon kasvattamiseksi. Tämä ei tarkoita sitä, että asiakkaan jokainen toive toteutettaisiin vaan pelkästään ne, jotka kasvattavat asiakkaan kokemaa arvoa suhteessa enemmän kuin kuluttavat aikaa ja rahaa. Asiakas voi olla myös sisäinen asiakas, jolloin tilaajan sekä toteuttavan tiimin tulisi toimia yhdessä, jotta ymmärretään, mitä tehdään ja kenelle. Päivittäisten rutiinien ongelmiin Lean startup-menetelmät auttavat siten, että turhia kokouksia ei tarvitse järjestää, koska tiimi on valtuutettu päätöksentekoon ja on myös kykeneväinen itse toteuttamaan päätetyt asiat. Pienten projektien kannalta pitää miettiä

Taulukko 18: Lean Startup-menetelmien hyödyt Soneran nykyisessä palvelukehityksessä

Lean Startup	
Ominaisuus	Poistuva haaste
Hypoteesiin perustuva strategia	Projekteissa ei varaa muutoksiin
Ongelmahaastattelut ja hypoteesien vahvistaminen asiakkailla	Hitaus ja speksaamisen kesto Asiakkaan rooli
Kehittäminen asiakkaan kanssa	Asiakkaan rooli Liian laajat projektitulaukset
Muutoksen ja epäonnistumisen odottaminen	Projekteissa ei varaa muutoksiin
Modernit mittarit: Asiakkaiden hankintakulu Asiakkaan elinkaariarvo Ratkaisun tuottama arvo	Mittaamista tehdään vähän, mutta mittareita ei seurata. Seurataan ainoastaan rahan ja ajan käyttöä.
Monialainen ja itseohjautuva tiimi joka tekee päätökset	allokaatioperusteinen tekeminen päättösvalta liian ylhäällä liikaa palavereita Siiloutuminen Teknologia - Liiketoiminta kuilu Tyytyväisyys tekemiseen
Nopea aloitus mahdollistaa nopean lopetuksen	resurssien vaje

että saadaanko monialainen osaava tiimi muodostettua. Yksinkertaista toteutusta tehtäessä kehitysmalli ei ole niin ratkaiseva[24].

Mittaaminen on yksi Lean Startup-menetelmän kivijaloista ja siinä on useampia systemaattisia keinoja lähestyä kehityksen mittaamista. Myös palvelukehityksen oppiminen on tärkeässä roolissa, sillä Startup-yritysten arvokkain etu on tiimin kyvykkyys. Hyviä palveluita ja ideoita löytyy riittämiin, mutta kykyä tehdä niistä ainutlaatuisia ja tuottava palveluita on vain harvoilla tiimeillä. Nykyiseen projektien rahoitusmalliin Lean Startup-malli ei sellaisenaan sovi, mutta sitä on jo sovellettu yrityksen muutamissa ketterissä hankkeissa. Projektien lopettaminen ei myöskään ole yhtä kallista tällä menetelmällä, sillä Lean Startup-mallissa rahaa ei käytetä vielä hankkeen alussa ennen kuin positiivisia signaaleita on löytynyt markkinoilta. Testiasiakkaita hankitaan useita ennen kuin toteutetaan mitään, mihin kuluisi suurta määrää rahaa. Asiakkaan rooli kasvaa valtavasti tämän kehitysmallin myötä ja asiakkaan osallistaminen on myös systemaattisesti toteutettu Lean Startup-menetelmissä. Uusien toimintamallien käyttöönotto myös vaatii sopivien roolien löytämistä niin kehittäjien kuin johtajienkin osalta. Tällä hetkellä kehittäjiä on Soneralla hyvin vä-

Taulukko 19: Scrum-menetelmien hyödyt Soneran nykyisessä palvelukehityksessä

Scrum	
Ominaisuus	Poistuva haaste
Sprintteihin perustuva tekeminen	Projekteissa ei varaa muutoksiin
Alussa luodaan visio, jonka jälkeen suunnittelut tähtäävät vain sprintin ajaksi	Hitaus ja speksaamisen kesto Asiakkaan rooli
Kehittäminen asiakkaan kanssa	Asiakkaan rooli
Projekti on pilkottu pienempiin tehtäviin	Liian laajat projektitilaukset
Muutoksen ja epäonnistumisen odottaminen	Projekteissa ei varaa muutoksiin
Suoritteiden mittaaminen rahan sijasta: Mitataan tehtävien suoritusnopeutta ja käyttäjäkertomusten tekemistä. Mitataan valmiin tuotteen arvoa asiakkaalle	Mittaamista tehdään vähän, mutta mittareita ei seurata. Seurataan ainoastaan rahan ja ajan käyttöä.
Monialainen ja itseohjautuva tiimi joka tekee päätökset	allokaatioperusteinen tekeminen päättösvalta liian ylhäällä liikaa palavereita Siiloutuminen Teknologia - Liiketoiminta kuilu Tyytyväisyys tekemiseen
Nopea aloitus mahdollistaa nopean lopetuksen	resurssien vaje

hän, joten tähän tehtävään ihmisten tulisi kouluttautua tai ihmisiä tulisi rekrytoida Soneran ulkopuolelta.

Perinteisten menetelmien kuten vesiputous- ja vaihe-portti-mallin hyödyt ovat kriittisten tuotteiden tekemisessä, jossa mahdollisuutta kokeiluun ei ole edes testiympäristön mielessä. Kriittisten järjestelmien suunnitteleminen etukäteen on tärkeää ja eri toiminnallisuuksien määrittely on hyvin ratkaisevaa. Mikäli toteuttava hanke on laajuudeltaan ja kokoluokaltaan pieni, ei ero ketteriin menetelmiin ole kovin suuri. Vesiputousmallissa on kuitenkin vapauksia tehdä monia samoja asioita kuin ketterissä menetelmissä. Nykyisen prosessin ollessa hyvin perinteisen kaltainen päästään vesiputousmallilla projektissa liikkeelle ilman työntekijöiden koulutusta. Kompleksisissa ratkaisuissa ja suurissa projekteissa vesiputousmallilla on hankala päästä lähellekään samoja lopputuloksia kuin ketterillä menetelmillä.

Skaalausmalleista sopivin Soneran kaltaiselle yritykselle olisi MAXOS, sillä se tukee pienempiä ydintiimejä, eikä vaadi Scrum-tiimin käyttöä. Pienempien tiimien vuoksi se aiheuttaa vähemmän resurssipulaa ja sen käyttöönotto ei vaadi yhtä suurta koulutusta kuin SAFe-mallin käyttäminen. MAXOS-skaalausmallissa ei ole pakotettuja aikaikkunoita, minkä vuoksi siinä ei kulu aikaa hukkaan. MAXOS sopii Lean

startup-tiimeille suoraan ja sen johtaminen on kevyempää, koska palavereita ja tapaamisia ei tarvitse järjestää etukäteen sovituin väliajoin, vaan tarpeen vaatiessa. Scrum-menetelmä sopii parhaiten ohjelmistokehitykseen, jota Soneralla tehdään itse varsin vähän, jonka takia Scrum-mallin ja SAFe-skaalausmallin käyttö ei ole niin hyödyllistä useissa osissa organisaatiota.

### 5.3 Tulevaisuuden tutkimuskohteet

Aihe on tällä hetkellä ajankohtainen, sillä Soneran organisaatiossa on otettu käyttöön Scrum-menetelmiä ja Lean Startup-menetelmiä. Näistä menetelmistä huolimatta suurin osa kehitysprojekteista tehdään vielä perinteisellä TSPromo-mallilla, joka mukailee vesiputous- tai vaihe-portti-mallia. Kun palvelukehitysprosessi on saatu kuvattua ja tästä on tehty laadullinen tutkimus, olisi luonnollinen siirtymä lähteä tutkimaan eri tapoja mitata palvelukehityksen tehokkuutta. Palvelukehitysprosessin tutkimisen lisäksi olisi myös hyvä ymmärtää muutosta, jonka organisaatio kokee yrittäessään muuttaa menetelmiään ketterämmiksi. Tästä aiheesta on tehty useita tutkimuksia, mutta niiden kohdistaminen Soneraan ja linkittäminen kehitysprosessin tehokkuuteen voitaisiin tutkia organisaation toimintatapojen muutoksen kannattavuutta. Tämän lisäksi tutkimuskohdetta voisi laajentaa useampaan yritykseen, jolloin voitaisiin tutkia samalla alalla toimivien yritysten palvelukehitystehokkuutta. Mikäli tästä saataisiin mitattavia tuloksia aikaiseksi, voitaisiin tutkimusta vielä laajentaa palvelukehitysprosessin tehokkuuden vaikutusta yrityksen kannattavuuteen ja tulokseen pidemmällä aikavälillä. Tutkimusta voisi myös viedä johtamisen suuntaan, ja tutkia palvelukehityksen tehokkuutta johtajasta ja johtamistyylistä riippuen. Tässäkin tutkimuksessa esitettiin johtajan tyyliä ja sen vaikutusta ketterien tiimien johtamiseen. Aiheesta voisii tutkia työntekijöiden tyytyväisyyttä ja mahdollistan rinnakkaisten tiimien suorituseroja suhteessa esimiehen johtamistyyliin.

## 6 Yhteenveto

Menestyksestä liiketoimintaa harjoittavan yrityksen on tarjottava hyviä palveluita tai tuotteita. Viimeisten vuosien aikana asiakkaan rooli palvelukehityksessä on noussut entistä tärkeämmäksi. Kansainvälistymisen myötä yhä useampi yritys tarjoaa palveluita, joiden tehtävänä on täyttää asiakkaiden muuttuvia tarpeita. Myös entistä nopeammin kehittyvä teknologia mahdollistaa monipuolisempia ja tyydyttävämpiä ratkaisuja asiakkaille. Tämän vuoksi palvelukehityksen haasteena on tuottaa entistä nopeammin entistä monipuolisempia ratkaisuja, jotka kuitenkin vähintään täyttävät ja mieluummin ylittävät asiakkaan odotukset. Samalla kun asiakkaiden tarpeet ovat muuttuneet haastavammiksi, usein tuotekehityskuluja haluttaisiin vähentää tai vähintään säilyttää ennallaan.

Tässä tutkimuksessa tavoitteena oli löytää hyötyjä eri palvelukehitysmalleista Soneran palvelukehitykseen. Tutkimuksessa paneuduttiin nykyisen palvelukehitysketjun kaikkiin osa-alueisiin. Tavoitteena oli löytää alan viimeisimmät kehitysmenetelmät ja suhteuttaa niitä nykyisiin menetelmiin. Työssä tutkittiin, miten palvelukehitystä tulisi johtaa ja minkälaisia vaatimuksia nämä uudet kehitysmenetelmät vaativat johtajilta.

Tutkimusta varten tehtiin 14 haastattelua, joihin osallistui henkilöitä eri yksiköistä. Kolmesta palvelukehitykseen liittyvästä yksiköstä haastateltiin neljää henkilöä, mutta liiketoimintayksiköstä haastateltiin kahta. Haastateltavina olivat yksiköiden johtajien lisäksi projektipäälliköitä ja kehittäjiä, jotta tutkimuskohteesta saatiin mahdollisimman paljon tietoa. Aihetta ei oltu tutkittu aikaisemmin, joten tutkimusmenetelmäksi valittiin tämän vuoksi haastattelututkimus.

Soneran teknologiayksikön projektikehitysmalli muistuttaa hyvin paljon perinteisiä kehitysmalleja, kuten vesiputous- ja vaihe-portti-mallia. Haastattelutulosten perusteella nähtiin, että ketterät mallit on tehty juuri lieventämään tai poistamaan näitä haasteita, joita perinteisissä malleissa ilmenee. Haasteita Soneran palvelukehityksessä syntyy IT-järjestelmien kompleksisuudesta, hitaasta etenemisestä, pitkistä esisuunnittelu ja määrittelyjaksoista. Tämän lisäksi haastattelututkimuksessa havaittiin kuilu liiketoiminnan ja hankkeita toteuttavan teknologiayksikön välillä. Myös päivittäistä tekemisistä löydettiin haasteita, kuten palaverien suuri määrä ja työn pirstaloituminen. Liiketoiminnassa ja projektiportfoliohallinnassa haasteita muodosti kehityskarttojen puutteet ja päätösten hitaus. Henkilöresurssien puute nousi yhtenä haasteena liiketoiminnan ja tuotteet ja palvelut-yksikön haastatteluissa.

Edellä mainitut haasteet pitäisi teorian mukaan ketterien ja asiakaslähtöisten menetelmien avulla lieventyä. Tässä työssä ei otettu kantaan siihen, mitä ketterien menetelmien käyttöönottoon tarvitaan, tai miten iso muutos vietäisiin läpi Soneran organisaatiossa. Ketterät ja Lean-menetelmät ovat luotu juuri perinteisten mallien korvaajiksi ja teoriaosuudessa löydetty havainnot tukevat haastattelututkimuksesta tehtyjä löytöjä. Suorituserot ketterillä ja perinteisillä menetelmillä huomioiden löydetty tulokset tukevat vahvasti sitä, että palvelukehitys ketterillä ja Lean-menetelmillä

tehostaa toimintaa. Palvelukehityksen tehostamisella tarkoitetaan asiakkaan kokeman arvon kasvamista, laadun paranemista ja budjetti sekä aikarajoitteiden pienenemistä suhteessa asiakkaan kokeman arvon kasvuun. Näiden lisäksi myös asiakastyytyväisyys sekä hankkeessa työskentelevien henkilöiden tyytyväisyys kasvavat siirryttäessä ketteriin ja asiakaslähtöisiin kehitysmalleihin kuten Lean Startup- tai Scrum-malleihin.



## Viitteet

- [1] D. Leffingwell, *Scaling software agility: best practices for large enterprises*. Pearson Education, 2007.
- [2] K. Conboy, “Agility from first principles: reconstructing the concept of agility in information systems development,” *Information Systems Research*, vol. 20, no. 3, pp. 329–354, 2009.
- [3] K. Beck, M. Beedle, A. Van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, *et al.*, “Manifesto for agile software development,” 2001.
- [4] S. Ambler, *Agile modeling: effective practices for extreme programming and the unified process*. John Wiley & Sons, 2002.
- [5] K. Beck, *Extreme Programming Explained*. Addison- Wesley, 2000.
- [6] J. Newkirk, “Introduction to agile processes and extreme programming,” in *Proceedings of the 24th international conference on Software engineering*, pp. 695–696, ACM, 2002.
- [7] H. Kniberg, *Scrum and XP from the Trenches: How we do Scrum*. Lulu. com, 2007.
- [8] K. S. Rubin, *Essential Scrum: A practical guide to the most popular Agile process*. Addison-Wesley, 2012.
- [9] A. Maglyas, U. Nikula, and K. Smolander, “Lean solutions to software product management problems,” *IEEE software*, no. 5, pp. 40–46, 2012.
- [10] J. P. Womack and D. T. Jones, *Lean thinking: banish waste and create wealth in your corporation*. Simon and Schuster, 2010.
- [11] M. Emiliani, “Improving business school courses by applying lean principles and practices,” *Quality Assurance in Education*, vol. 12, no. 4, pp. 175–187, 2004.
- [12] A. David. J, “Kanban: Successful evolutionary change for your technology business,” WA: *Blue Hole Press*, 2010.
- [13] M. O. Ahmad, J. Markkula, and M. Oivo, “Kanban in software development: A systematic literature review,” in *Software Engineering and Advanced Applications (SEAA), 2013 39th EUROMICRO Conference on*, pp. 9–16, IEEE, 2013.
- [14] M. Sahota, “An agile adoption and transformation survival guide: working with organizational culture,” *InfoQ*, <http://www.infoq.com/minibooks/agile-adoption-transformation>, 2012.
- [15] E. Ries, *The lean startup: How today’s entrepreneurs use continuous innovation to create radically successful businesses*. Crown Books, 2011.

- [16] S. Blank, "Why the lean start-up changes everything," *Harvard Business Review*, vol. 91, no. 5, pp. 63–72, 2013.
- [17] R. G. Cooper, "Stage-gate systems: a new tool for managing new products," *Business horizons*, vol. 33, no. 3, pp. 44–54, 1990.
- [18] A. . H. Booz, "New product management for the 1980s," *New York*, 1982.
- [19] C. Larman and V. R. Basili, "Iterative and incremental development: A brief history," *Computer*, no. 6, pp. 47–56, 2003.
- [20] N. M. A. Munassar and A. Govardhan, "A comparison between five models of software engineering," *IJCSI*, vol. 5, pp. 95–101, 2010.
- [21] S. Nerur, R. Mahapatra, and G. Mangalaraj, "Challenges of migrating to agile methodologies," *Communications of the ACM*, vol. 48, no. 5, pp. 72–78, 2005.
- [22] V. Vinekar, C. W. Slinkman, and S. Nerur, "Can agile and traditional systems development approaches coexist? an ambidextrous view," *Information systems management*, vol. 23, no. 3, pp. 31–42, 2006.
- [23] P. Sfetsos and I. Stamelos, "Empirical studies on quality in agile practices: A systematic literature review," in *Quality of Information and Communications Technology (QUATIC), 2010 Seventh International Conference on the*, pp. 44–53, IEEE, 2010.
- [24] <http://www.infoq.com/articles/standish-chaos> 2015, "Chaos report 2015," 2015.
- [25] M. Leeuwen, "Agile scaling@ topicus: scaling scrum with help of agile scaling frameworks at topicus finance," 2015.
- [26] A. Vaidya, "Does dad know best, is it better to do less or just be safe? adapting scaling agile practices into the enterprise," *PNSQC. ORG*, pp. 1–18, 2014.
- [27] C. Larman and B. Vodde, *Practices for scaling lean & Agile development: large, multisite, and offshore product development with large-scale scrum*. Pearson Education, 2010.
- [28] D. Leffingwell, *Agile software requirements: lean requirements practices for teams, programs, and the enterprise*. Addison-Wesley Professional, 2010.
- [29] A. Singleton, *Unblock! A Guide to the New Continuous Agile: Release More Frequently to Fly Past Your Competitors*. Assembla, Incorporated, 2013, 2013.
- [30] P. Sydänmaanlakka, "Älykäs johtajuus," *Helsinki: Talentum*, 2004.
- [31] N. A. Bonner, "Predicting leadership success in agile environments: An inquiring systems approach," *Academy of Information and Management Sciences Journal*, vol. 13, no. 2, p. 83, 2010.
- [32] J. Kulla *et al.*, "'käskyttämällä ei pitkälle pääse." tutkimus suurten suomalais-yritysten toimitusjohtajien johtamiskokemuksista," 2011.

- [33] M. E. Doyle and M. K. Smith, *Born and bred?: Leadership, heart and informal education*. YMCA George Williams College, 1999.
- [34] J. Nikkanen, “Akateeminen johtajuus muutoksessa—tutkimus helsingin yliopiston ylimmän johdon esimies- ja vuorovaikutusvalmennuksesta,” *Helsingin yliopiston hallinnon julkaisuja*, vol. 74, 2011.
- [35] B. Boehm, “Get ready for agile methods, with care,” *Computer*, vol. 35, no. 1, pp. 64–69, 2002.
- [36] J. Schiel, *The ScrumMaster Study Guide*. CRC Press, 2011.
- [37] J. T. Hakala, “Menetelmällisiä koetuksia,” *Teoksessa Aaltola, J. & Valli, R.(toim.) Ikkunoita tutkimusmetodeihin*, vol. 1, pp. 10–23, 2001.
- [38] P. Alasuutari, *Laadullinen tutkimus*. Vastapaino, 1999.
- [39] S. Hirsjärvi and H. Hurme, *Tutkimushaastattelu: teemahaastattelun teoria ja käytäntö*. Gaudeamus Helsinki University Press, 2008.
- [40] J. Eskola and J. Suoranta, *Johdatus laadulliseen tutkimukseen*. Vastapaino, 1998.
- [41] J. Highsmith, *Agile project management: creating innovative products*. Pearson Education, 2009.
- [42] I. Otava-Keskinen, “Varma it vakuuttaa,” *Tivi*, 2012.
- [43] J.-P. Kevätsalo *et al.*, “Innovaatiokäytänteiden muutos siirryttäessä perinteisistä kehitysmalleista ketterään kehittämiseen. case iphone-sovelluksen kehitys finanssialan yrityksessä,” 2011.